修 士 論 文 の 和 文 要 旨

研究科・専攻	大学院情報システム学研究科情報ネットワークシステム学専攻博士前期課程		
氏　　名	金　燕	学籍番号	1352007
論 文 題 目	TCP バージョンの違いによる BitTorrent の性能分析		

要　　旨

近年、P2P ネットワーク技術を用いたアプリケーションが非常に多く利用されている。その中で、BitTorrent は人気のある代表的なファイル共有 P2P アプリケーションである。P2P アプリケーションは、ピアがクライアントである同時にサーバになり、コンテンツ配信が行われ、サーバにかかる負荷を分散できる。また、ピアはコンテンツをダウンロードする同時にアップロードを行い、ピアの帯域幅を有効に利用できる。一方、P2P アプリケーションはインターネットトポロジーを考慮せずに設計され、オーバーレイネットワークを形成する。そのため、複数の AS(Autonomous System)を横断する AS 間のトラヒックが生成され、70%の BitTorrent のダウンロードが AS 間で行われるという研究報告もある。このような冗長なトラヒックの増加により、ピアのダウンロード速度が低下する問題があるため、AS 間のトラヒックを減らす必要がある。そこで、本論文では、BitTorrent において TCP バージョンの違いによる性能分析を行う。BitTorrent の性能は自身の戦略だけではなく、TCP 輻輳制御にも関係がある。AS 内の伝搬遅延を異なる場合、帯域幅が異なる場合及びピアの非チョーク数が異なる場合 TCP バージョンごとに分析する。それらの結果により、TCP バージョンごとに BitTorrent の性能が違う原因とローカライゼーションへの影響を明らかにし、相互に与える影響を分析する。

伝搬遅延を 1ms と 5ms に設定した場合、TCP Vegas は最初にシーダから非チョークされてないピアは、AS 外から持ってきたピースを他の TCP 輻輳制御アルゴリズムより AS 内で積極的にダウンロードするため、ローカライゼーション度合いも高くなる。伝搬遅延が 5ms の場合 BIC-TCP はシーダから大量のピースをダウンロードしたピア数が少なくなり、その AS 内のピース数が少なくなったため、ローカライゼーション度合いが悪くなる。BIC-TCP の再送回数が多くなるため、スループットが遅くなる。帯域幅が 10Mbps の場合 TCP Westwood は、シーダから選ばれたピアの AS において、ピース数が多いため、AS 内から積極的にピースをダウンロードするが、ピースが少ない AS のピアは AS 内と AS 外の帯域幅条件が同一になるため、帯域幅が 100Mbps の場合より、AS 外からのダウンロードが増え、ローカライゼーション度合いが悪くなる。また、帯域幅が 10Mbps の場合、BIC-TCP の再送回数がやや少なくなり、帯域幅が 100Mbps の場合よりスループットが少し速くなる。非チョーク数が 10 の場合、TCP Illinois はシーダから非チョークされたピア数が増え、AS 内のピース数が多くなる。そして、シーダから非チョークされてないピアは AS 外からダウンロードしたピースを非チョーク数が 3 の場合より積極的にダウンロードしたため、ローカライゼーション度合いが高くなる。また、非チョーク数が 10 の場合、全ての TCP 輻輳制御アルゴリズムの再送回数が少なくなるが、ピア平均スループットは TCP Vegas のスループットがやや速くなる。非チョーク数が 10 の場合、TCP New Reno は再送回数が少なくなるが、スループットが速い TCP フローがなくなる。

平成２６年度修士論文

TCPバージョンの違いによるBitTorrentの性能分析

大学院情報システム学研究科情報ネットワークシステム学専攻

学 籍 番 号：1352007

氏　　名：金　燕

主任指導教員：大坐畠　智　准教授

指 導 教 員：加藤　聰彦　教授

指 導 教 員：長岡　浩司　教授

提 出 年 月 日：平成２７年１月２６日（月）

概要

近年、P2P ネットワーク技術を用いたアプリケーションが非常に多く利用されている。VoIP (Voice Over IP)や高画質テレビ番組放送等のアプリケーションは、数千万人のユーザにサービスを提供している。その中でも、BitTorrent は依然として人気のある代表的なファイル共有 P2P アプリケーションである。2014 年上半期の BitTorrent トラフィックはアジア太平洋のインターネットトラヒックの 26.95%を占めている。P2P アプリケーションは、ピアがクライアントである同時にサーバになる。ピアがサーバになり、コンテンツ配信が行われると、クライアント/サーバモデルの問題点であるサーバにかかる負荷を分散することができる。また、ピアはコンテンツをダウンロードする同時に、アップロード帯域幅を利用してアップロードを行い、他のピアにサービスを提供する。このような方式はピアの帯域幅を有効に利用できる。一方、P2P アプリケーションはインターネットトポロジーを考慮せずに設計されているため、オーバーレイネットワークを形成する。そのため、複数の AS(Autonomous System)を横断する AS 間のトラヒックが生成され、70%の BitTorrent のダウンロードが AS 間で行われるという研究報告もある。このような冗長なトラヒックの増加により、ピアのダウンロード速度の低下を引き起こす問題があるため、AS 間のトラヒックを減らす必要がある。そこで、本論文では、BitTorrent において、TCP バージョンの違いによる性能分析を行う。BitTorrent は様々な戦略でダルンロードを行うが、主に Rarest-first 戦略と Tit-for-tat 戦略を用いる。BitTorrent の性能は BitTorrent 自身の戦略だけではなく、TCP 輻輳制御にも関係している。AS 内の伝搬遅延を異なる場合、AS 内の帯域幅が異なる場合及びピアの非チョーク数が異なる場合の BitTorrent 性能を TCP 輻輳制御アルゴリズムごとに分析を行う。それらの結果により、TCP バージョンごとに BitTorrent の性能が違う原因とローカライゼーションへの影響を明らかにし、相互に与える影響を分析する。

目次

図目次

表目次

第 1 章　序論

1.1 はじめに

近年、様々なインターネットアプリケーションの中でP2Pネットワーク技術を用いたアプリケーションが非常に多く利用されている。例えば、VoIP (Voice Over IP)や高画質テレビ番組放送等のアプリケーションは、数千万人のユーザにサービスを提供している。その中でも、BitTorrentは依然として人気のある代表的なファイル共有P2Pアプリケーションである。2014 年上半期のBitTorrentはアジア太平洋のインターネットトラヒックの26.95%を占めている[1]。

P2Pアプリケーションは、ピアがクライアントである同時にサーバになる。これはクライアント/サーバモデルと違うところである。ピアがサーバになり、コンテンツ配信が行われると、クライアント/サーバモデルの問題点であるサーバにかかる負荷を分散することができる。また、ピアはコンテンツをダウンロードする同時に、アップロードを行い、他のピアにサービスを提供する。このような方式はピアのアップロード帯域幅を有効に利用できる。一方、P2Pアプリケーションは課題として、インターネットトポロジーを考慮せずに設計されているため、オーバーレイネットワークを形成する。そのため、複数のAS(Autonomous System)を横断するAS間のトラヒックが生成され、70%のBitTorrentのダウンロードがAS間で行われるという研究もある[2]。このような冗長なトラヒックの増加により、ピアのダウンロード速度の低下を引き起こす問題がある。それため、ピアのダウンロード速度を低下するAS間のトラヒックを減らす必要がある。

本論文では、AS 間のトラヒックを減らすため、BitTorrent において、TCP バージョンの違いによる性能分析を行う。BitTorrent は P2P ネットワーク技術を用いたアプリケーションの中で最も人気のあるファイル共有アプリケーションの 1 つである[3]。BitTorrent は様々な戦略でダウンロードを行うが、主に Rarest-first 戦略と Tit-for-tat 戦略を用いる。BitTorrent の性能は BitTorrent 自身の戦略だけではなく、TCP 輻輳制御にも関係している。それで、本論文では AS 内の伝搬遅延を異なる場合、AS 内の帯域幅が異なる場合、ピアの非チョーク数が異なる場合の BitTorrent 性能を TCP 輻輳制御アルゴリズムごとに分析を行う。それらの結果により、TCP バージョンごとに BitTorrent の性能が違う原因とローカライゼーションへの影響を明らかにし、相互に与える影響を分析する。

1.2 本論文の構成

本論文は以下のように構成する。第 2 章では BitTorrent プロトコル、TCP 輻輳制御アルゴリズムおよび AS について説明する。第 3 章では本論文で行う実験環境、実験トポロジーおよび測定項目について述べる。第 4 では実験結果を示し、その結果について分析を行う。第 5 章では、実験をまとめする。

第 2 章　BitTorrent プロトコルと関連技術

本章では、本研究に使用した P2P アプリケーションである BitTorrent の概要と、本研究で用いた 5 つの TCP 輻輳制御アルゴリズムについて説明する。2.1 節では BitTorrent プロトコルの概要について述べる。2.2 節では TCP 輻輳制御アルゴリズムについて述べる。2.3 節では AS について述べ、2.4 節では関連技術について述べる。

2.1 BitTorrent の概要

BitTorrent はファイル共有のためのプロトコルおよびアプリケーションである。BitTorrent は P2P ネットワーク技術を用いるため、ピア同士でダウンロードおよびアップロードを行う。BitTorrent はファイルを複数のピースに分割し、ピースごとでダウンロードが行われる。ファイルを分割してダウンロードを行うと、負荷を少なくなり、ダウンロード速度が高くなる。

2.1.1 BitTorrent の動作概要

図 1 は BitTorrent の動作概要[4]を示す。BitTorrent の動作は主に 3 つのステップで説明できる。

1. クライアントはダウンロードする際に Web サーバから Torrent ファイルを取得する。この Torrent ファイルはダウンロードしたいファイルのハッシュ値、ファイル名、ファイルサイズ、ピースの長さ、トラッカーの URL などの情報を所持している。この Torrent ファイルを用いてダウンロードを行う。
2. Torrent ファイルを取得したクライアントは、トラッカーへアクセスし、ダウンロードの開始とピアリストを要求する。トラッカーとは、クライアントから要求に応じ、ピアの IP アドレスとポートの情報を更新して返すサーバ側のプログラムである。トラッカーはダウンロードを開始したピアをスウォーム（同じ Torrent ファイルをダウンロード/アップロードするピア全体または一部）内のピアとして登録し、ピアのファイル取得状況やアップロード/ダウンロード容量を所持している。また、定期的にピアにピアリストを提供する。
3. トラッカーからピアリストを取得したクライアントはファイルのダウンロードを開始する。BitTorrent はファイルを一定サイズ(128KB~1MB)のピースに分割し、ピア間でダウンロードを行う。ファイルのピースをすべて所持しているピアをシーダと言い、シーダはアップロードだけを行う。また、ファイルのピースを一部所持しているピアをリーチャと言い、リーチャはダウンロードとアップロードを行う。

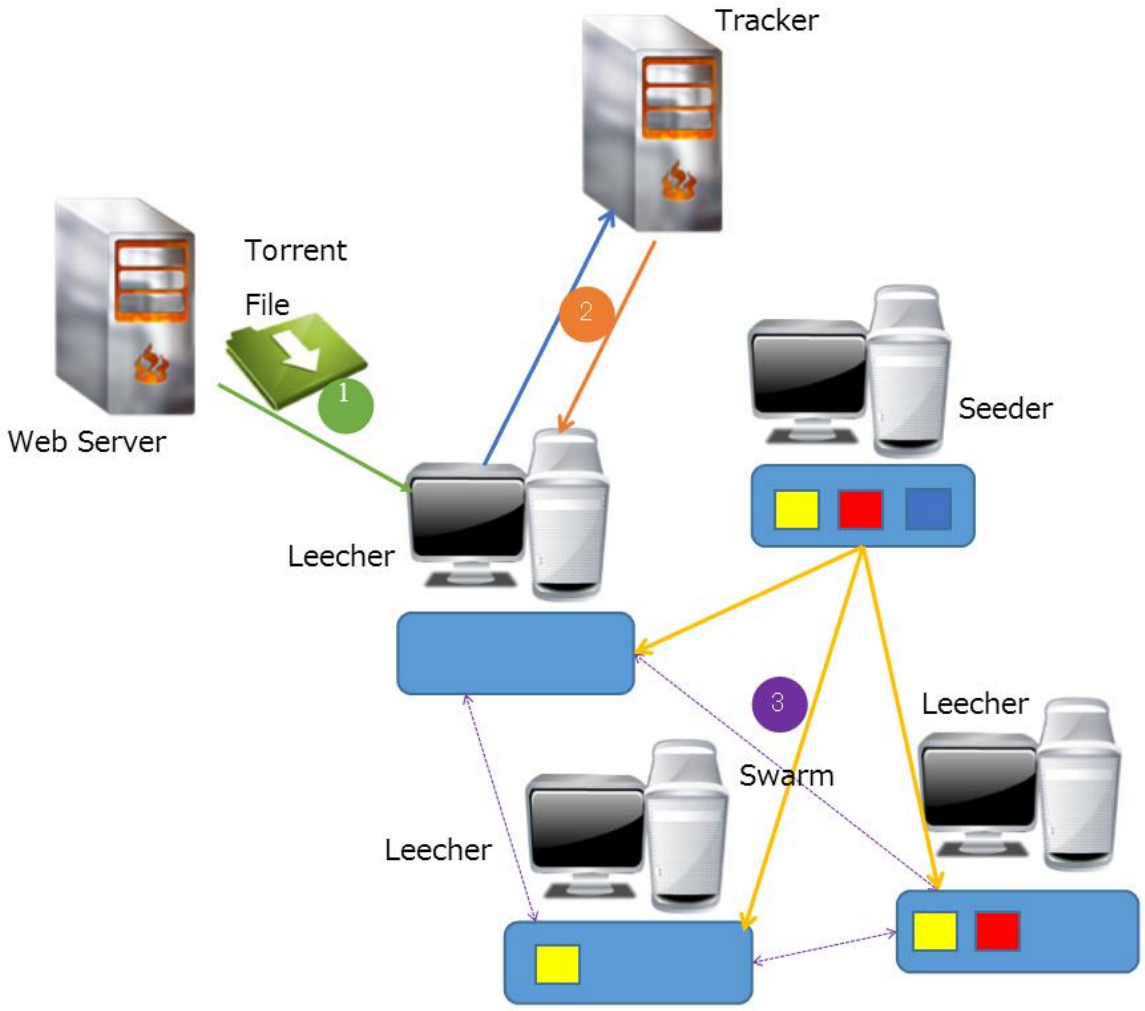

図 1　BitTorrent の動作概要

2.1.2 ピース選択戦略

BitTorrent のピース選択戦略には、Random-first 戦略、Rarest-first 戦略、End-Game-Mode 戦略がある。

クライアントは起動して最初に Random-first 戦略[4]を用い、ランダムでダウンロードするピースを選択する。

BitTorrent は基本的に Rarest-first 戦略[4]でダウンロードを行う。Rarest-first 戦略とは複数のピアからピースをダウンロードする際に、それぞれのピアが所持しているピースの中で最も少ないピースをダウンロードしようとする戦略である。このような戦略は、あるピアがスウォームから離脱してしまった場合、そのピアしか所持していないピースがあると、そのピースをダウンロードできなくなることを防ぐことができる。

それ以外にも、End-Game-Mode 戦略[4]という戦略がある。End-Game-Mode 戦略はファイルの最後のピースをダウンロードする時に用いる。帯域幅が低いピアから最後のピースをダウン

ロードすることになると、ダウンロード時間が長くなる可能性がある。この問題を避けるために、最後のピースのダウンロード要求をピアリストのすべてのピアに送信し、複数のピアから同時にダウンロードする。どのピアからは関係なく、最後のピースをダウンロードすると、他のダウンロードを終了させる。

2.1.3 ピア選択戦略

BitTorrent のピア選択戦略には 2 つある。1 つは Tit-for-tat 戦略で、もう 1 つは楽観的非チョーク戦略である。主に用いる戦略は Tit-for-tat 戦略である。

Tit-for-tat 戦略[4]ではピアのダウンロード速度によって非チョーク（相手のピアのダウンロードを許可すること）するピアを決定する。このダウンロード速度は最近 20 秒間のダウンロード量の平均を指標として、高い順 3 つを非チョークする。チョーク（相手のピアのダウンロードを許可しないこと）、非チョークの状態が頻繁に変化すると、ダウンロード速度が低下するため、Tit-for-tat 戦略では 10 秒ごとに非チョークするピアを決定する。

ダウンロード速度だけではより良いピア選択を行うことができないため、楽観的非チョーク戦略[4]ではランダムにピアを選択する。楽観的非チョーク戦略は 30 秒ごとに非チョークするピアを決定する。

2.2 TCP 輻輳制御アルゴリズム

TCP 輻輳制御アルゴリズムはバージョンによって様々存在する。TCP 輻輳制御アルゴリズムは大きく分けると 3 種類があり、それぞれ Loss-based 方式、Delay-based 方式と Hybrid 方式である。この 3 つの方式はパケット廃棄が発生した時に、輻輳ウィンドウサイズの変化が異なり、輻輳制御方式も違う。以下に 3 つの方式[5]を説明する。

・Loss-based 方式

Loss-based 方式はパケット廃棄が発生するまで輻輳ウィンドウを増加させる。パケット廃棄が発生すると、輻輳ウィンドウを減少させる。この方式はパケット廃棄を輻輳発生の指標として用いるため、パケット廃棄を回避することができない。

・Delay-based 方式

Delay-based 方式はパケット廃棄が発生する前に、各パケットの RTT(Round Trip Time)を監視し、輻輳ウィンドウの増加の指標として、輻輳回避を行う。この方式は RTT の増加を輻輳発生の指標として用いるため、パケット廃棄を避けることができる。

・Hybrid 方式

Loss-based 方式と Delay-based 方式を組み合わせた方式である。最初は輻輳ウィンドウを大幅に増加させる。Delay-based 方式のように RTT を輻輳ウィンドウの増加の指標とするため、パケット廃棄が発生する前に RTT が増加し始めると RTT の増加量を減少させる。パケット廃棄が発生したら、Loss-based 方式と同様に輻輳ウィンドウを減らす。つまり、輻輳の発生までは

Delay-based 方式を用い、輻輳が発生する時は Loss-based 方式を用いる。

2.2.1 TCP New Reno

TCP New Reno 輻輳制御アルゴリズム[5]は 3 つの方式の中で Loss-based 方式になる。TCP New Reno はスロースタートと輻輳回避アルゴリズムで構成させる。それぞれのアルゴリズムは輻輳ウィンドウサイズ（cwnd）を増加する速度が異なる。

スロースタートアルゴリズムでは 1 つの ACK を受信するごとに輻輳ウィンドウサイズを 1 パケット増加させる。輻輳ウィンドウ値を Slow Start Thresh(ssthresh)の値まで増加する。輻輳回避アルゴリズムでは 1 つの ACK を受信するごとに輻輳ウィンドウサイズをその逆数の値だけ増加させる。そのアルゴリズムは以下のように表す。

$$cwnd = \begin{cases} cwnd + 1 \ (cwnd < ssthresh) \\ cwnd + 1/cwnd \ \ (cwnd \geq ssthresh) \end{cases} \quad (1)$$

ここで$cwnd$は輻輳ウィンドウサイズ、$ssthresh$はスロースタートしきい値を表している。

一方、輻輳が発生する場合には、次の式のように輻輳ウィンドウサイズを減少させる。

$$cwnd = \begin{cases} cwnd/2 \ (重複ACKの場合) \\ 1 \ (タイムアウトの場合) \end{cases} \quad (2)$$

すなわち、1 つの重複 ACK を 3 回受信した場合、$ssthresh$の値を輻輳ウィンドウの半分まで減少させ、タイムアウトの場合、輻輳ウィンドウの値は 1 にする。図 2 は TCP New Reno 輻輳制御アルゴリズムの輻輳ウィンドウの時間による変化を示す。

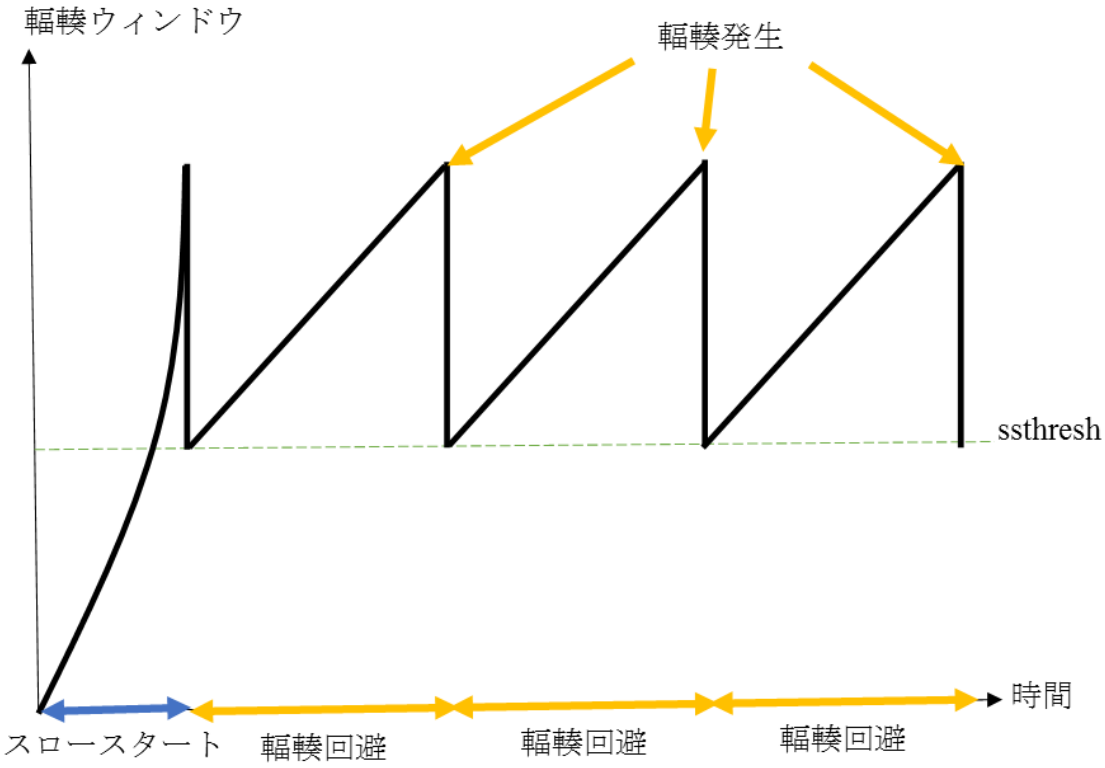

図 2　TCP New Reno の輻輳ウィンドウの時間による変化

2.2.2 BIC-TCP

BIC(Binary Increase Congestion Control) TCP 輻輳制御アルゴリズム[6]は Loss-based 方式である。BIC-TCP は TCP New Reno と同様にスロースタートと輻輳回避アルゴリズムで構成されている。BIC-TCP 輻輳制御アルゴリズムの輻輳ウィンドウの時間による変化を図 3 に示す。最初にはスロースタートを行い、パケット廃棄が発生するまで輻輳ウィンドウを増加させる。輻輳が発生すると、輻輳ウィンドウの増加の仕方が変化する。

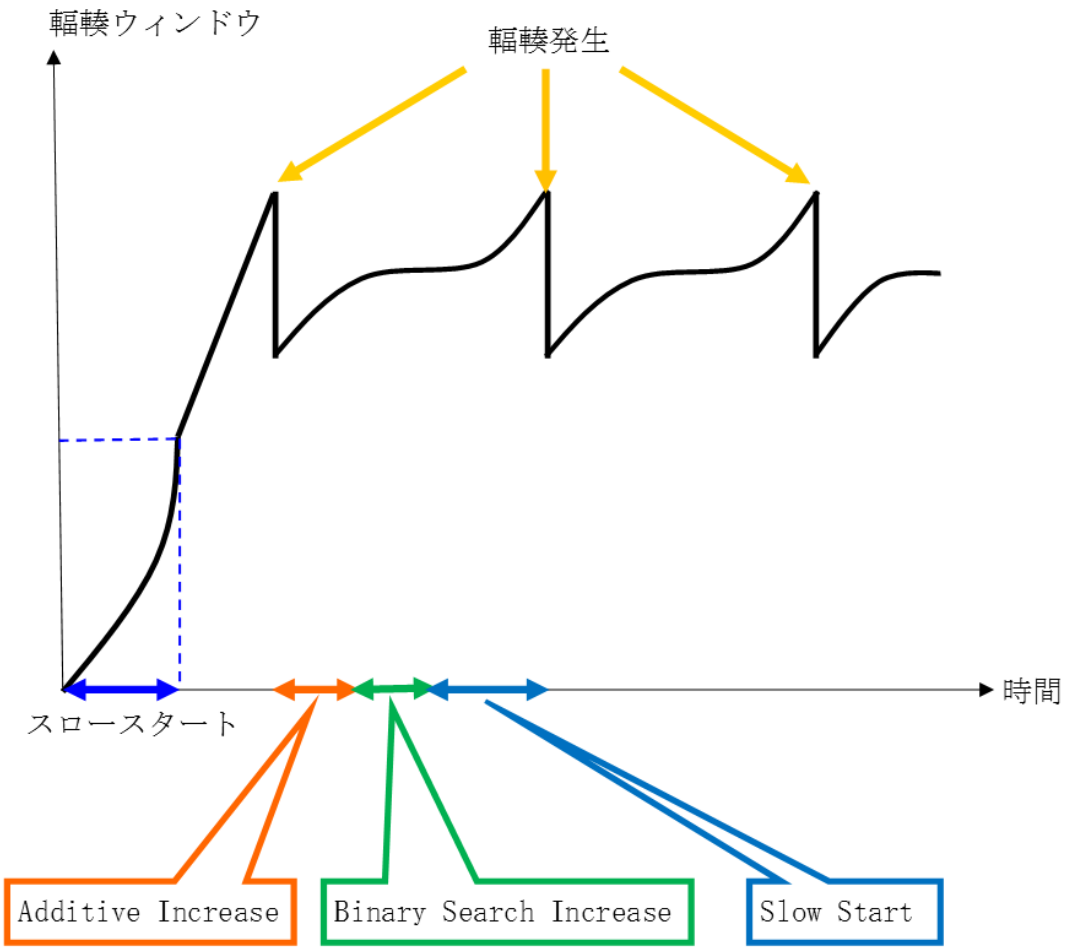

図 3　BIC-TCP の輻輳ウィンドウの時間による変化

図 3 によると、BIC-TCP は Additive Increase 段階に入り、輻輳ウィンドウを増加させる。パケット廃棄が発生する時に、パケット廃棄直前の輻輳ウィンドウの値をW_{max}、パケット廃棄直後の輻輳ウィンドウの値をW_{min}とする。そして、W_{max}とW_{min}の中間値を求める。Additive Increase 段階では輻輳ウィンドウの値を最大増加量S_{max}だけ増加する。式は以下のようになる。

$$cwnd = cwnd + S_{max}/cwnd \quad (3)$$

輻輳ウィンドウがW_{max}とW_{min}の中間値との差がS_{max}以下になると、Additive Increase 段階を離れ、Binary Search Increase 段階に入る。

Binary Search Increase 段階では、W_{max}とW_{min}の中間値から輻輳ウィンドウの増加量を求める。輻輳ウィンドウの増加の式は以下のようになる。

$$cwnd = cwnd + (\frac{W_{max}+W_{min}}{2} - cwnd)/cwnd \quad (4)$$

輻輳ウィンドウがW_{max}とW_{min}の中間値に到達した後、W_{min}の値を現在の輻輳ウィンドウの値に変更し、W_{max}との中間値を再び求め、輻輳ウィンドウを増加させる。このプロセスを何回繰

り返し、輻輳ウィンドウがW_{max}に到達すると、スロースタート段階に入る。

スロースタート段階では新たに輻輳ウィンドウの最大値として$W_{max}+S_{min}$にする。輻輳ウィンドウの増加量は以下の式のようになる。

$$cwnd = cwnd + S_{min}/cwnd \quad (5)$$

輻輳ウィンドウの最大値になる後に、次は、$W_{max}+2*S_{min}$ 、$W_{max}+4*S_{min}$ 、$W_{max}+8*S_{min}$ 、… 、$W_{max}+S_{max}$のように変化する。

輻輳が発生する場合、輻輳ウィンドウを減少させる。BIC-TCP では RTT の公平性を考えており、複数のフローが存在する場合、以下の式のように輻輳ウィンドウを減少させる。βは固定値である。

$$cwnd = cwnd*(1-\beta) \quad (6)$$

2.2.3 TCP Westwood

TCP Westwood 輻輳制御アルゴリズム[7][8]は Loss-based 方式である。TCP New Reno と同様にスロースタートと輻輳回避アルゴリズムで構成されている。図 4 は TCP Westwood 輻輳制御アルゴリズムの輻輳ウィンドウの時間による変化を示している。図 4 に示したように、TCP Westwood は、まず、スロースタート段階に入り、パケット廃棄が発生するまで、輻輳ウィンドウを増加させる。輻輳発生を検出する場合、輻輳ウィンドウの値を以下の式のように変化させる。

重複 ACK を 3 回受信した場合

$$ssthresh = \frac{BWE \times RTT_{min}}{seg\_size} \quad (7)$$

$$cwnd = ssthresh \quad (8)$$

タイムアウトの場合

$$ssthresh = \frac{BWE \times RTT_{min}}{seg\_size} \quad (9)$$

$$ssthresh = 2 \ (ssthresh < 2) \quad (10)$$

$$cwnd = 1 \quad (11)$$

式で、BWEは利用可能が帯域幅の推定値を表し、RTT_{min^n}は測定された最小の RTT であり、seg_sizeはセグメントサイズを表している。

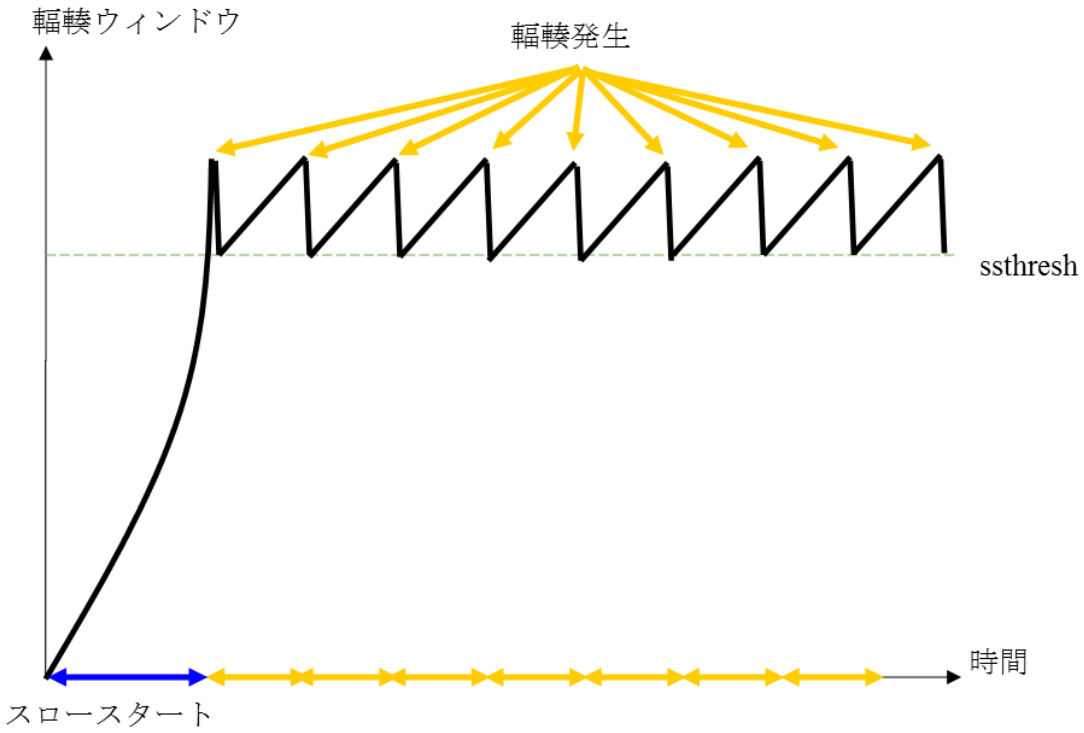

図 4　TCP Westwood の輻輳ウィンドウの時間による変化

2.2.4 TCP Vegas

TCP Vegas 輻輳制御アルゴリズム[9]は 3 つの方式の中で Delay-based 方式である。TCP Vegas は RTT に基づいて、推測したスループットと測定したスループットを用いて輻輳制御を行う。推測したスループットと測定したスループットの計算式は以下のようになる。

$$diff = Expected - Actual = \frac{cwnd}{basedRTT} - \frac{cwnd}{RTT} \quad (12)$$

$cwnd$は現在の輻輳ウィンドウ、$basedRTT$は最小の RTT を表し、RTTは一個前の RTT 値を表す。上記の 2 つの式を用い、以下の式のように輻輳ウィンドウを増減させる。

$$cwnd = \begin{cases} cwnd + 1 \, (diff < \alpha) \\ cwnd \, (\alpha < diff < \beta) \\ cwnd - 1 \, (diff > \beta) \end{cases} \quad (13)$$

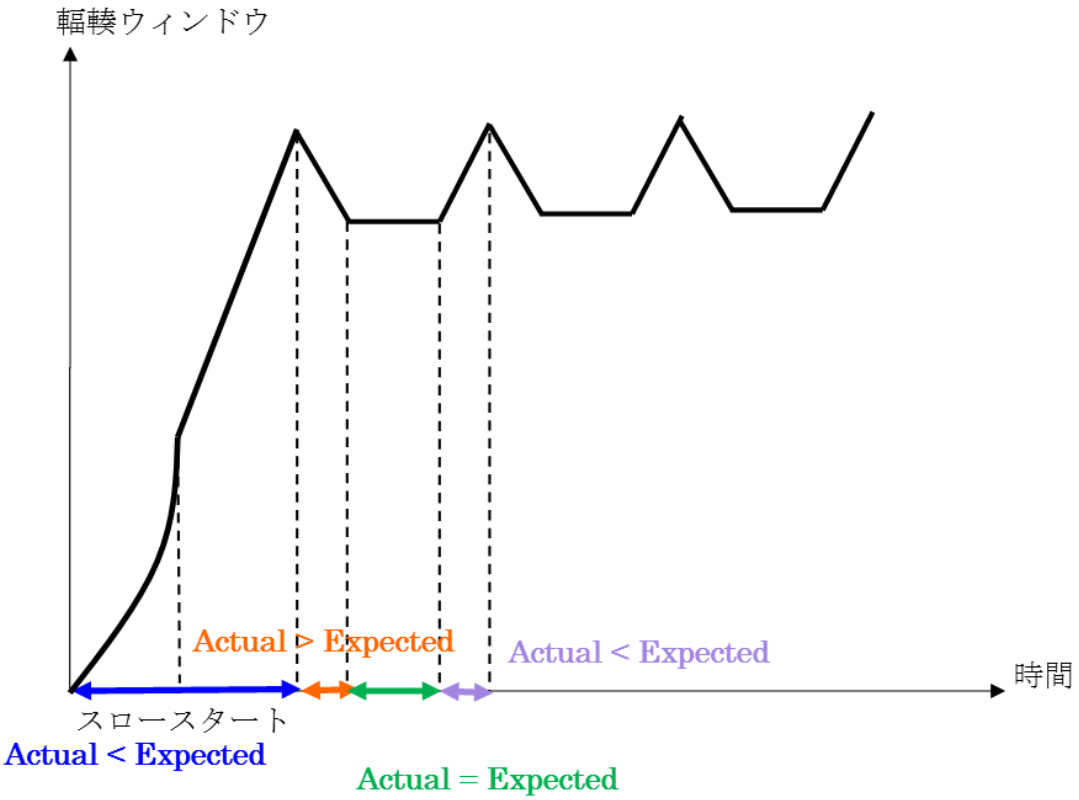

図 5　TCP Vegas の輻輳ウィンドウの時間による変化

図 5 に示すように TCP Vegas は、始めはスロースタートを用いて増加する。$Actual < Expected$になる時点でスロースタートから切り替える。その後、上記の式のように、$Actual > Expected$の場合には、輻輳ウィンドウサイズを 1 減らし、$Actual < Expected$の場合には、輻輳ウィンドウサイズを 1 増やす。そして、$Actual = Expected$の場合には、輻輳ウィンドウサイズを変更しなく、その値を維持する。

2.2.5 TCP Illinois

TCP Illinois 輻輳制御アルゴリズム[10][11]は Loss-based 方式と Delay-based 方式を合わせた Hybrid 方式である。TCP Illinois は図 6 に示すように、最初にはスロースタートを行う。その後、RTT の値により輻輳ウィンドウを増減させる。

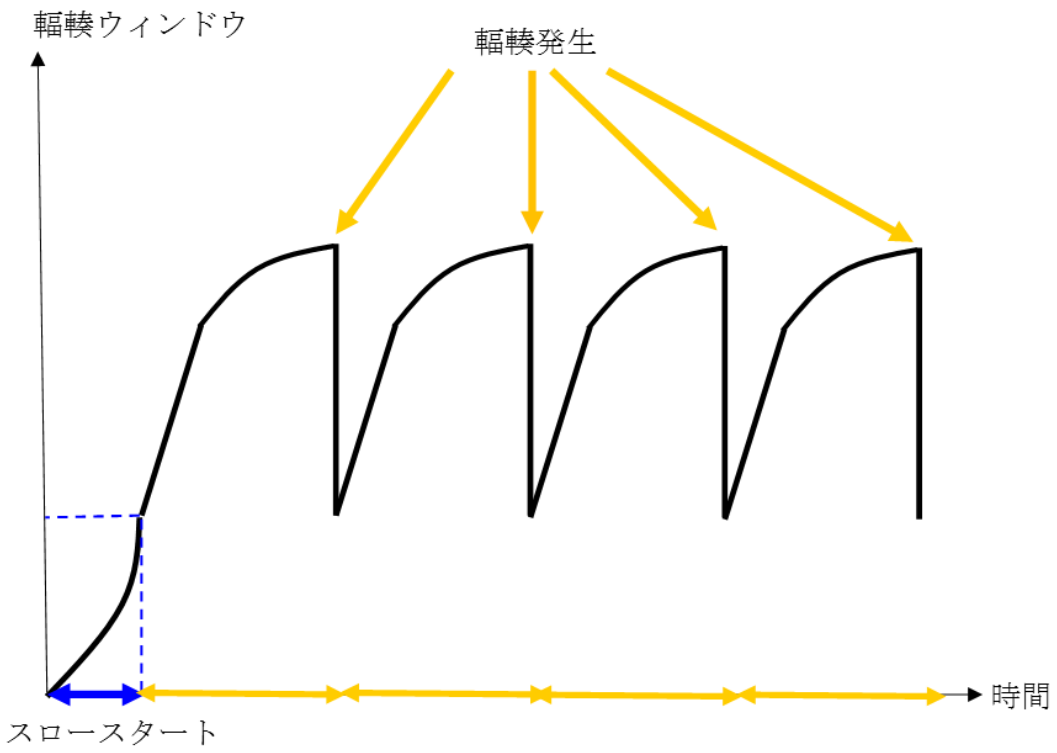

図 6　TCP Illinois の輻輳ウィンドウの時間による変化

まず、輻輳ウィンドウの増減の値を決めるパラメーターについて説明する。

$$d_m = T_{max} - T_{min} \quad (14)$$

$$d_a = T_a - T_{min} \quad (15)$$

上記の式で、d_mは最大キューイング遅延時間、d_aは平均キューイング遅延時間を表す。T_{max}は RTT の最大値、T_aは RTT の平均値、T_{min}は RTT の最小値を表す。

$$\alpha = f_1(d_a) = \begin{cases} \alpha_{max} \ (d_a < d_1) \\ \frac{k_1}{k_2 + d_a} \ (otherwise) \end{cases} \quad (16)$$

$$\beta = f_2(d_a) = \begin{cases} \beta_{min} \ (d_a < d_2) \\ k_3 + k_4 d_a \ (d_2 < d_a < d_3) \\ \beta_{max} \ (otherwise) \end{cases} \quad (17)$$

上記の式で、d_1 、d_2 、d_3は図 7 に示されているように、α曲線とβ曲線によって決まる。

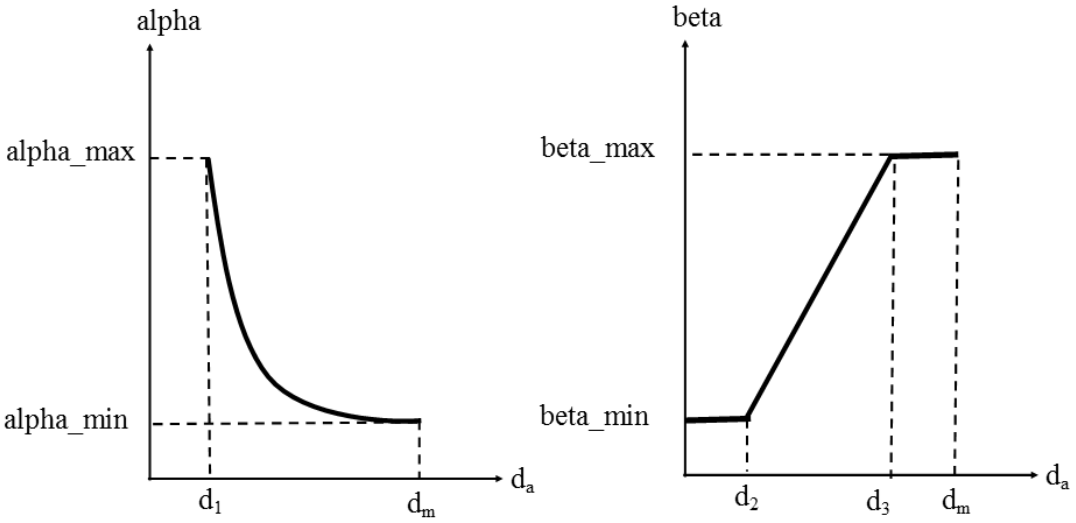

図 7　α曲線、β曲線　vs.　d_a

また、k_1～k_4は以下の式により求める。

$$k_1 = \frac{(d_m - d_1)\alpha_{min}\alpha_{max}}{\alpha_{max} - \alpha_{min}} \quad \text{and} \quad k_2 = \frac{(d_m - d_1)\alpha_{min}}{\alpha_{max} - \alpha_{min}} - d_1 \quad (18)$$

$$k_3 = \frac{\beta_{min} d_3 - \beta_{max} d_2}{d_3 - d_2} \quad \text{and} \quad k_4 = \frac{\beta_{max} - \beta_{min}}{d_3 - d_2} \quad (19)$$

T_{max}またはT_{min}が更新されるとk_1～k_4の値が更新される。αとβの値は RTT の変化とともに変更する。TCP Illinois はα値を用い、以下の式のように輻輳ウィンドウを増加させる。

$$cwnd = cwnd + \alpha / cwnd \quad (20)$$

また、輻輳が発生し、重複 ACK を受信した場合、以下の式を用い、輻輳ウィンドウを減少させる。

$$cwnd = cwnd - \beta * cwnd \quad (21)$$

2.3 AS (Autonomous System)

AS (Autonomous System)というのはインターネットにつながる一つまたは複数のルーティングポリシー配下にある IP ネットワークやルーターの集合のことである。例えば、大企業、大学、会社の一部門、または企業グループをそれぞれ一つの AS としてネットワークを構成する。各 AS

には 32 ビットの番号が割り当てられるが、それを AS 番号 (Autonomous System Number)と呼ぶ。AS 番号は IANA (Internet Assigned Numbers Authority)により、IP アドレスとともにブロック単位で RIR (Regional Internet Registry)に割り当てられている。日本では担当 RIR である APNIC (Asia-Pacific Network Information Centre)か、日本の NIR (National Internet Registry)である JPNIC (Japan Network Information Center)に申請し、承認を得る。

AS は 3 種類に分けられる。それぞれ、バックボーン AS (Backbone AS)、スタブ AS (Stub AS)およびトランジット AS (Transit AS)である。以下に詳細に説明する。

・ バックボーン AS

バックボーン AS は図 8 に示すように、一番上の層の AS である

・ スタブ AS

スタブ AS は図 8 に最下層の AS であり、1 つ AS のみ接続されている AS である。

・ トランジット AS

スタブ AS を繋がっている AS をトランジット AS と呼ぶ。トランジット AS は 2 つの AS 間のトラヒックを処理し、サービスを提供するため、無料ピアリングもあるが、有料になる場合もある。

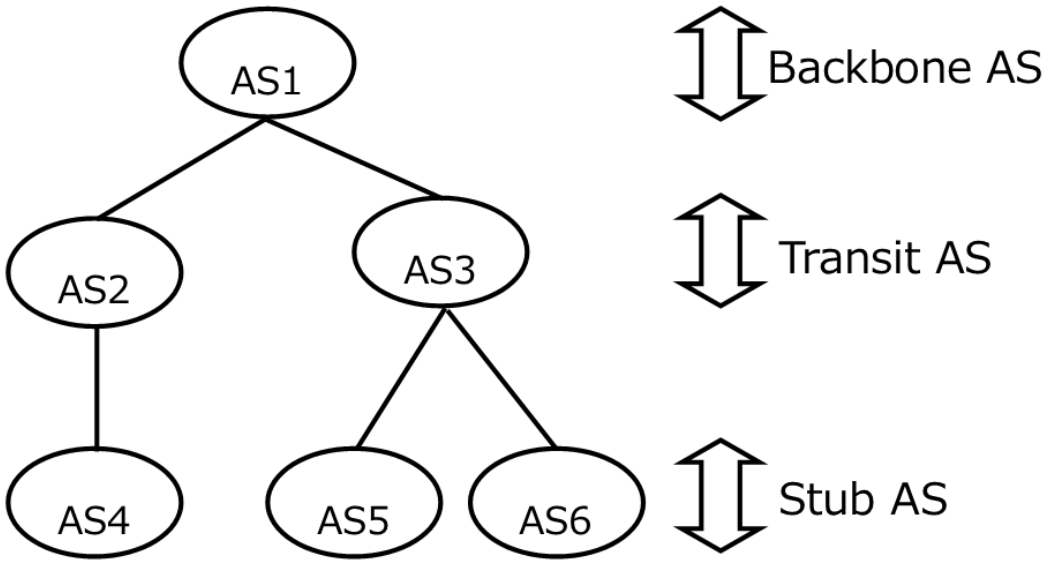

図 8　3 種類の AS

インターネット上では、このように様々な AS が存在し、すべての AS がつながっているわけではないため、AS 間のトラヒックが生成される。実際に、BitTorrent の場合、70%のダウンロードが AS 間で行われるという研究もある[2]。バックボーンルータートランジット AS は非常に高いコストがかかり、スタブ AS も接続されていない AS との通信は必ずトランジット AS を経路するため、コストがかかる。また、AS 間の通信は物理的な距離が遠いため、遅延が長くなり、通信速度は低くなる問題もある。

2.4 関連研究

2.4.1 輻輳制御アルゴリズムの性能評価の関連研究

文献[12]では LEDBAT (Low Extra Delay Background Transport)という輻輳制御アルゴリズムを紹介した。LEDBAT は最近 BitTorrent が導入した新たな輻輳制御アルゴリズムである。LEDBAT は以下の 2 つの改善を目指す。一つ目は、キューイングによる追加遅延を制限し、ユーザトラヒックの余りが同じアクセスボトルネックを共有することの干渉を減らす。二つ目は、効率的に利用可能なリンク容量を使用し、同時にユーザに良好な BitTorrent 性能を提供する。LEDBAT と既存の TCP 輻輳制御アルゴリズムを比較する実験を通して、一部の目的は達成したが、ある実験では低い優先度を示す。それは、LEDBAT は TCP の設定に依存しているためである。

LEDBAT は Delay-based 方式に分けられ、文献[13]では、他の Delay-based 方式の TCP 輻輳制御アルゴリズム (NICE、VEGAS と LP)との性能比較を行った。その結果、LEDBAT はキューイング遅延が他の TCP より短く、ボトルネックリンクの帯域幅を効率的に利用できる。

文献[14]では、TCP Westwood+、TCP New Reno と TCP Vegas において、その性能を評価し、比較を行った。その結果、TCP New Reno と TCP Westwood+のプロトコル間の友好性があるため、TCP Vegas はそれらと共存する場合、帯域幅シェアをつかむことができない。また、帯域幅割当におけるプロトコル内の公平性は TCP New Reno より TCP Westwood+が良い。損失の多いリンク利用率の提供は TCP Westwood+が TCP New Reno より改善されている。

そこで、本研究では、TCP 輻輳制御アルゴリズムの性能の違いと、その性能による BitTorrent の性能を分析し、BitTorrent に最適な TCP 制御アルゴリズムを考察する。

2.4.2 ローカライゼーションの関連研究

文献[15]では、現存する P2P ネットワークの AS 間トラヒックを削減するために、AS トポロジーを用いる手法を提案した。本文献では、BitTorrent において、ピア選択やピース選択を行う時に、相手のピアの位置を意識するように修正し、実験を行った。その結果、AS 間のトラヒックが減少させ、ダウンロード時間も減らした。

文献[16]では、AS 間のトラヒックを削減するための手法として、3 つを提案した。(1) ピアによって直近に交換されるファイルのピースの予測、(2) キャッシングアルゴリズム、(3) ヒット率向上のための SDN 経路制御を提案した。特に、レアピースが最初に交換される Rarest-first 戦略が BitTorrent に採用されているため、透過的にピアにとってレアなピースの予測をする手法が効果的であると考え、BITFIELD/HAVE メッセージを網内で検知する。プログラマブルノード上で予測した網内におけるレアピースと各ピアが保持するレアピースの一致度をシミュレーショ

ンにより評価した。その結果、ピアが観測する上位約 33%のレアピースを平均 40%のピアに対して 70%以上の確率で予測できる。また、上位 10%のレアピースは、平均 30%のピアに対して、70%以上の確率で予測できる。つまり、提案手法で網内の観測のみだけて直近で交換されるピースを優先的にキャッシュし。キャッシュのヒット率を向上できる。

そこで、本研究では、アプリケーション層のネットワークトポロジーとトランスポート層の TCP 制御アルゴリズムおよび BitTorrent 側のパラメーターに着目し、TCP バージョンごとに BitTorrent のピア AS 間のピース取得状況やその速度を分析する。

2.4.3 BitTorrent における TCP バージョンの違いによる影響のクロスレイヤ測定分析

文献[17]では、BitTorrent 自身の戦略と TCP 輻輳制御に対して、クロスレイヤ測定分析することで、BitTorrent において、TCP バージョンごとのローカライゼーション度合いとスループットを調査した。その結果、BIC-TCP が他の TCP バージョンより高いスループットを維持している。また、同じ AS から取得したピース数も BIC-TCP が比較的に多く、ローカライゼーション度合いも他の TCP バージョンより高い結果になっている。そして、シーダが AS 外に置いている場合、スループットとローカライゼーション度合いが全体的に下がることがわかる。

そこで、本研究では、様々なパラメーターを用い、複数の AS のトポロジーを作り、各 TCP バージョンによる BitTorrent のローカライゼーション度合いとスループットをさらに詳細的に分析する。

第３章　実験環境と測定項目

この節には、本論文の実験を行われた実験環境と実験トポロジーおよび測定項目について説明する。3.1 節では実験環境の概要について述べ、3.2 節では実験のトポロジーを紹介し、3.3 節では測定項目であるスループット、3.4 節ではローカライゼーション度合いに関して説明する。

3.1 実験環境

本論文では、AS 間のトラヒックを減らすため、BitTorrent において、TCP バージョンの違いによる性能分析を行う。性能評価はネットワークシミュレーターである ns-3[18]上で行う。ns-3 には TcpSocketBase[19]の子クラスとして様々な TCP 輻輳制御アルゴリズムが実装されている。本実験に使用した TCP 輻輳制御アルゴリズムは表 3 に示している。以下の 5 種類の TCP 輻輳制御アルゴリズムの中で TCP New Reno と TCP Westwood だけ ns-3 に実装されている。そのため、本実験では Network Simulation Cradle (NSC)[19]を用いる。NSC は実 OS で実装されたネットワークのプロトコルスタックをシミュレータされたネットワークで使用できるようにするフレームワークである。BitTorrent は文献[20]で提案した ns-3 のための BitTorrent モデルを用いる。本モデルには brite[21]を用い、トポロジー変更を行う。

表 1　TCP 輻輳制御アルゴリズム

TCP 輻輳制御アルゴリズム	方式
New Reno	Loss-based
BIC	Loss-based
Westwood	Loss-based
Vegas	Delay-based
Illinois	Hybrid

3.2 トポロジー

本実験ではトラッカー1 ピア、シーダ 1 ピアとリーチャ 16 ピアを用意した。また、BitTorrent の AS 間のピースダウンロード状況を調査するために、リーチャを複数の AS に設定する。トラッカーとシーダは AS1 に設定する。様々なトポロジーとパラメーターを用い、評価を行う。

本実験でダウンロードするファイルの情報は表 2 に示している。

表 2　ファイル情報

ファイルサイズ	ピースサイズ	ピース数
100MB	64KB	1600

本実験の基本トポロジーは図 10 に示している。

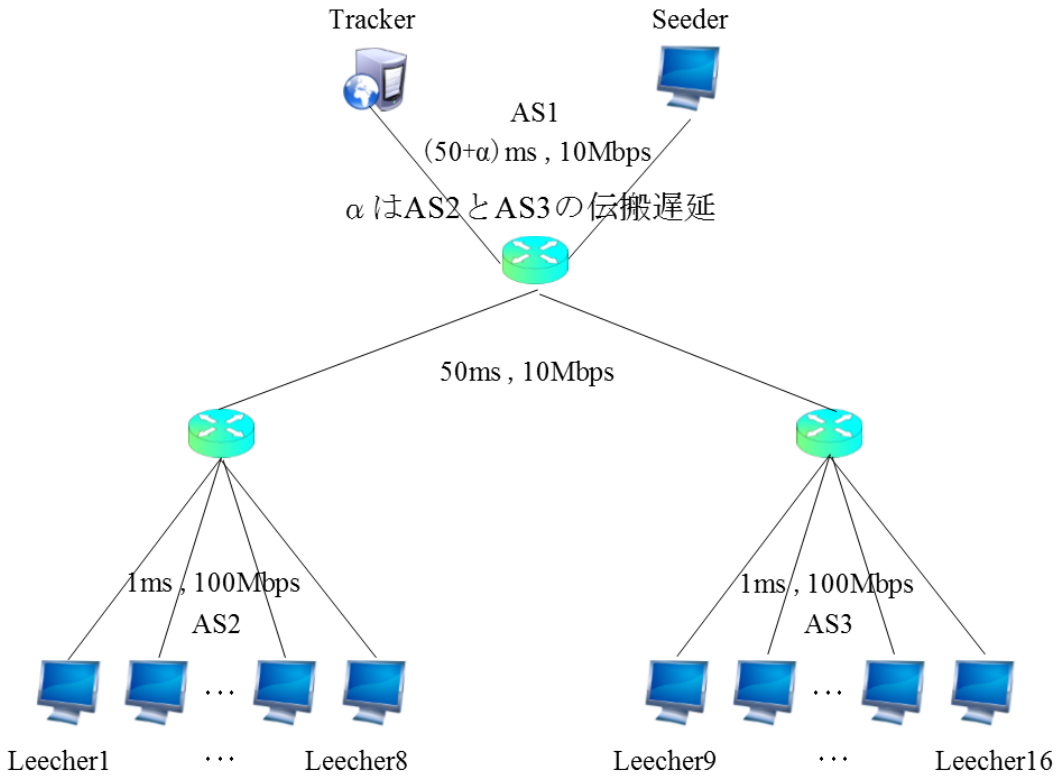

図 9　実験のトポロジー

上記のトポロジーのピアの詳細な情報は表 3 に示している。

表 3　ピア情報

ピア	IP アドレス	AS
トラッカー	10.1.0.1	1
シーダ	10.2.0.41	1
リーチャ 1 ～8	10.2.0.43~57	2
リーチャ 9~16	10.2.0.59~73	3

トポロジーのパラメーターは表 4 のように変更し、評価を行う。

表 4　トポロジーパラメーター

AS 内の伝搬遅延	1ms、5ms

AS 内の帯域幅	100Mbps、10Mbps
ピアの非チョーク数	3、10

3.3 評価メトリック

本論文の第 1 章に述べたように、P2P ネットワーク技術は下位の IP ネットワーク資源を意識せず、IP ネットワークのトポロジーを考慮せずに通信が行われている。そのため、複数の AS を横断する AS 間のトラヒックを生成する。この問題を考察するため、本実験では、複数の AS を持つネットワーク環境を作り、BitTorrent のピアが AS 内のピアと AS 外のピアからのピース取得状況を分析する。分析を通して、各 TCP 輻輳制御アルゴリズムのローカライゼーション度合いを計算する。計算式は以下に示す。

$$ローカライゼーション度合い=\frac{AS内のピアから取得したピース数}{総ピース数} \quad (22)$$

AS 内と AS 外のトラヒックをより詳細に分析するために以下のようにファイルを構成するためにダウンロードしたピースの平均ホップ数を計算する。

$$\frac{\sum_i chunk_i \times hops}{File\ Size} \quad (23)$$

TCP 輻輳制御アルゴリズムの違いにより、輻輳制御の仕組みが異なり、輻輳ウィンドウの増減する方式が異なる。そのため、BitTorrent のクライアントがファイルをダウンロードする際に、ダウンロード速度に影響を与える。本実験では、各 TCP 輻輳制御アルゴリズムにおいて、BitTorrent のダウンロード速度を調査するために、その指標として各ピアの平均スループットを測定する。

第４章　実験結果と分析

この節には、前章で述べた実験環境で行った実験の結果を示し、その分析を行う。4.1 節では、AS 内の遅延を変更した場合のスループットとローカライゼーション度合いの結果を示し、分析を行う。4.2 節では、AS 内の帯域幅を変更した場合のスループットとローカライゼーション度合いの結果を示し、分析を行う。4.3 節では、シーダとリーチャの非チョーク数を変更し、実験の結果を比較する。

4.1 AS 内の遅延が異なる場合

本実験では、AS 内の伝搬遅延が異なる場合、TCP バージョンごとにそのスループットとローカライゼーション度合いを比較する。3 章で述べた基本トポロジーの AS 内の伝搬遅延を 1ms と 5ms に変更し、その結果を考察する。

4.1.1 ローカライゼーション度合い測定の結果

図 10 と図 11 は伝搬遅延が 1ms と 5ms の場合のローカライゼーション度合いの累積分布関数を示す。図 12 と図 13 は伝搬遅延が 1ms と 5ms の場合のダウンロードしたピースの平均ホップ数の累積分布関数を示す。両方の場合とも TCP Vegas のローカライゼーション度合いが良い。図 14 から図 23 に示された各ピアのピース取得状況のマトリックスを見ると、最初にシーダから非チョークされてないピアは、AS 外から持ってきたピースを他の TCP 輻輳制御アルゴリズムより AS 内で積極的にダウンロードする。それで、TCP Vegas のダウンロードしたピースの平均ホップ数が少なく、ローカライゼーション度合いも高くなる。伝搬遅延が 5ms の場合 BIC-TCP のローカライゼーション度合いが悪くなる。伝搬遅延が 5ms の場合、シーダから大量のピースをダウンロードしたピア数が少なくなり、その AS 内のピース数が少なくなったため、ローカライゼーション度合いが悪くなり、ダウンロードしたピースの平均ホップ数も大きくなる。

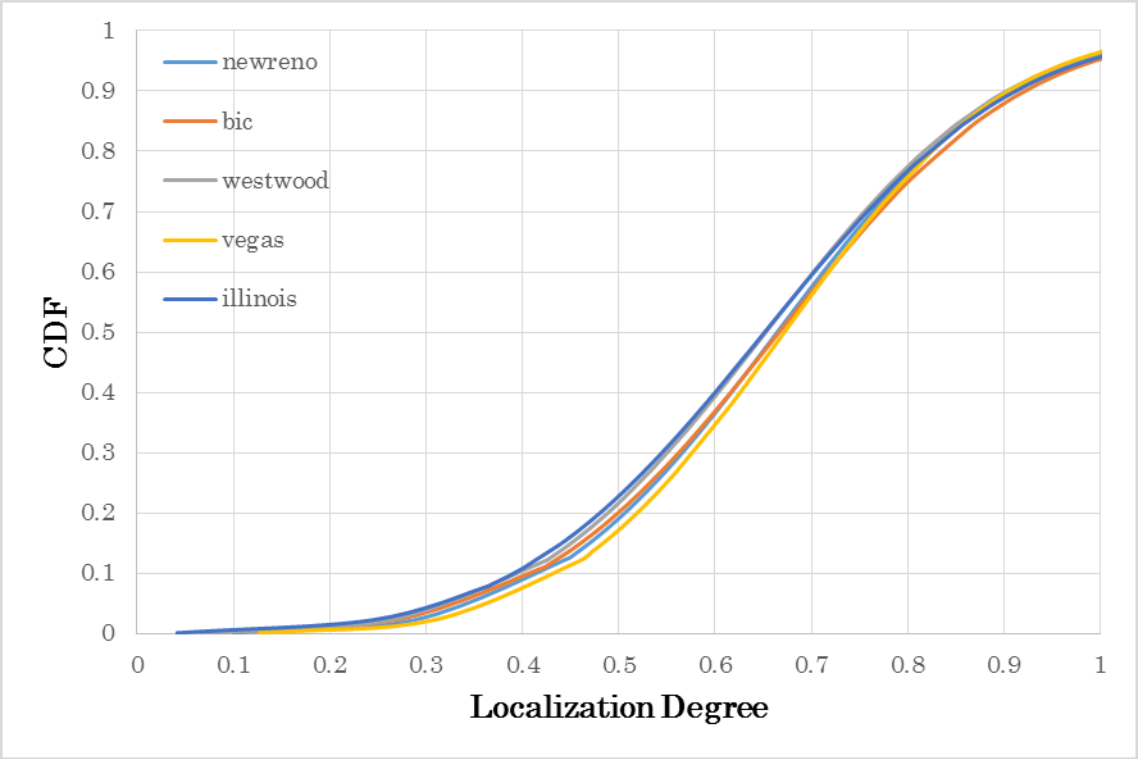

図 10　伝搬遅延が 1ms の場合ローカライゼーション度合い累積分布関数

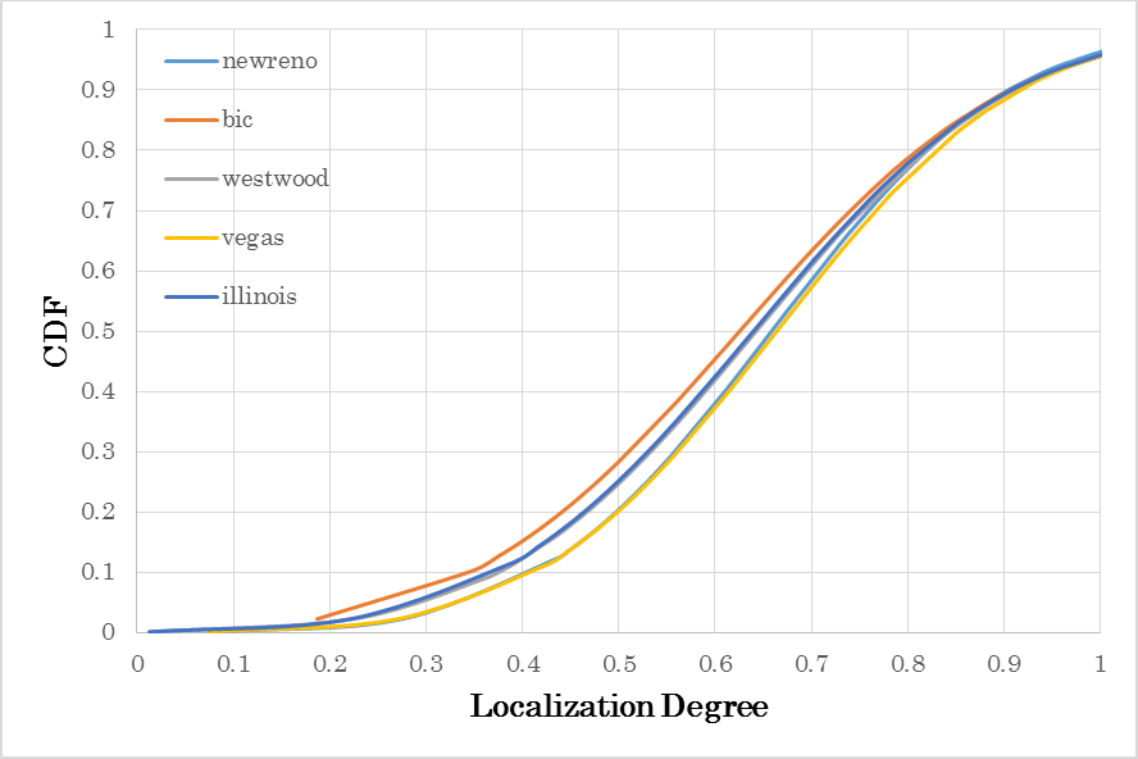

図 11　伝搬遅延が 5ms の場合ローカライゼーション度合い累積分布関数

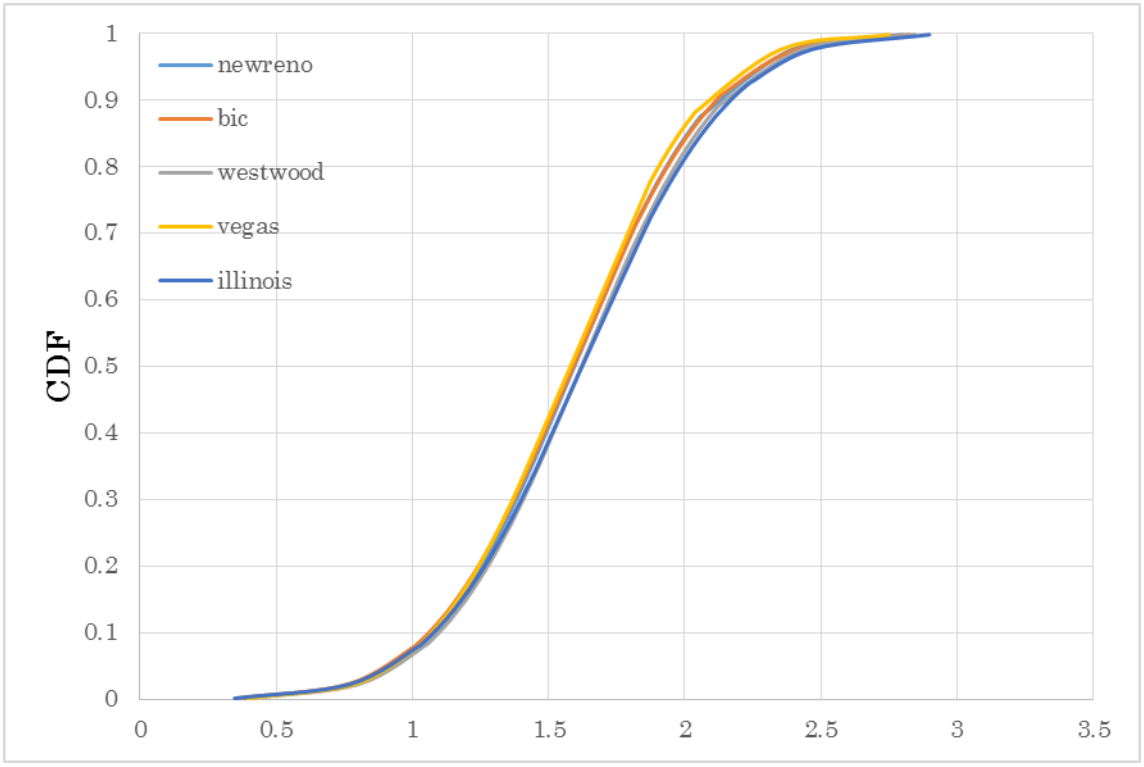

図 12　伝搬遅延が 1ms の場合ダウンロードしたピースの平均ホップ数累積分布関数

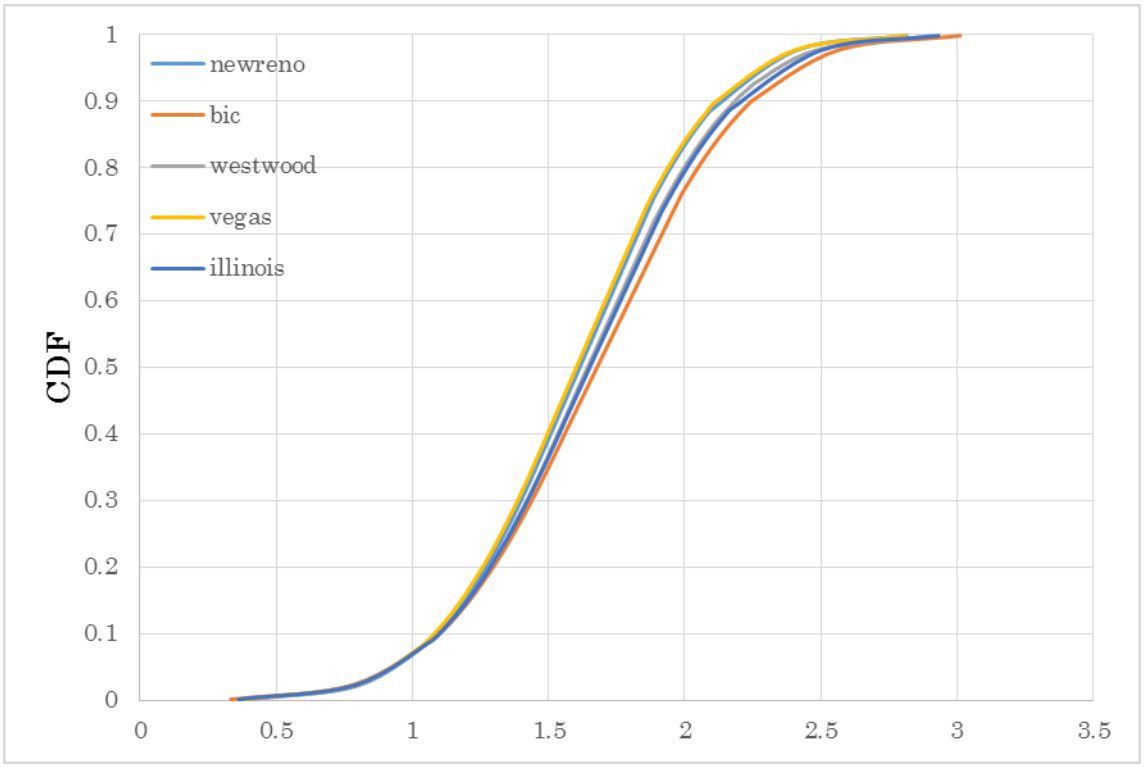

図 13　伝搬遅延が 5ms の場合ダウンロードしたピースの平均ホップ数累積分布関数

図 14 から図 23 は伝搬遅延が 1ms と 5ms の場合、TCP 輻輳制御アルゴリズムごとの各ピアのピース取得状況を示している。縦軸は送信側のピアの IP アドレスであり、横軸は受信側のピアの IP アドレスである。横軸の IP アドレスの”10.2.0”は省略する。IP アドレスが”10.2.0.41”のピアはシーダである。表の数値は各ピアからダウンロードしたピース数を示している。紫色の部分は AS 内ピアからダウンロードしたピース数を表し、緑色は AS 外からの取得ピース数を表している。

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	15	45	0	35	25	1	30	0	333	611	0	492	0	209	0	248
10.2.0.43		106	116	109	108	122	116	113	6	0	3	5	1	1	2	4
10.2.0.45	114		120	113	101	113	123	99	18	18	18	20	16	19	19	16
10.2.0.47	112	110		106	102	110	115	112	2	0	4	4	4	0	2	3
10.2.0.49	125	126	139		128	130	116	129	12	14	19	8	14	16	10	13
10.2.0.51	134	111	140	120		135	132	136	7	5	9	7	9	8	9	7
10.2.0.53	120	108	131	122	118		120	118	2	0	1	1	3	1	1	0
10.2.0.55	82	94	93	88	89	100		99	14	16	12	15	16	15	13	15
10.2.0.57	82	77	86	78	91	79	76		4	0	2	0	1	0	0	0
10.2.0.59	118	126	71	96	88	97	98	112		312	297	315	252	303	215	217
10.2.0.61	101	111	98	109	95	93	104	108	561		578	540	498	561	609	436
10.2.0.63	91	101	100	100	106	97	107	104	0	8		0	34	3	42	42
10.2.0.65	95	104	102	103	106	91	96	94	427	409	434		456	430	440	450
10.2.0.67	98	94	99	95	119	112	101	95	1	1	1	0		1	1	2
10.2.0.69	112	95	90	96	105	101	91	94	4	2	2	0	4		4	19
10.2.0.71	114	100	101	102	96	104	94	99	5	0	2	2	65	10		110
10.2.0.73	106	94	96	96	98	104	93	93	193	179	204	171	209	208	206	

図 14　伝搬遅延が 1ms の場合 TCP New Reno の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	553	481	55	102	0	2	4	516	0	21	0	36	171	0	20	39
10.2.0.43		470	431	390	279	198	191	488	98	91	59	70	68	66	103	64
10.2.0.45	394		412	442	370	253	366	413	93	148	91	70	41	90	89	67
10.2.0.47	44	44		0	71	164	136	0	111	113	124	98	78	85	91	72
10.2.0.49	92	97	109		188	240	256	98	94	102	102	122	95	80	81	127
10.2.0.51	0	0	4	57		27	28	2	105	110	86	110	105	85	91	123
10.2.0.53	0	0	2	15	2		4	14	83	110	117	103	119	98	100	111
10.2.0.55	0	0	11	17	15	28		10	99	93	119	100	103	130	85	114
10.2.0.57	452	446	510	517	617	629	558		101	98	115	94	98	116	117	91
10.2.0.59	4	6	1	0	4	2	0	4		61	78	77	82	78	66	62
10.2.0.61	3	2	4	9	8	2	6	6	141		131	142	160	136	142	144
10.2.0.63	2	2	13	2	0	7	9	7	115	118		95	123	114	121	119
10.2.0.65	14	1	9	14	12	9	3	0	116	102	109		117	134	128	151
10.2.0.67	5	0	2	6	4	5	1	8	110	109	109	101		128	119	105
10.2.0.69	9	2	0	1	3	0	0	8	90	86	94	94	90		83	93
10.2.0.71	8	7	9	0	1	9	8	12	127	122	121	126	164	141		136
10.2.0.73	0	22	8	1	13	4	8	2	149	174	151	171	154	146	169	

図 15 伝搬遅延が 1ms の場合 BIC-TCP の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	546	532	0	0	447	0	158	3	71	0	20	0	23	24	32	16
10.2.0.43		499	468	262	510	524	247	324	98	108	116	75	108	80	95	110
10.2.0.45	462		424	413	436	472	533	548	104	108	102	82	89	82	97	98
10.2.0.47	0	0		18	0	24	31	23	105	103	104	100	83	90	96	94
10.2.0.49	2	0	1		0	1	0	1	96	97	104	118	88	92	108	100
10.2.0.51	381	377	403	429		358	388	386	101	96	86	111	92	114	102	99
10.2.0.53	2	0	51	148	2		141	105	92	87	99	110	117	102	95	115
10.2.0.55	107	90	145	204	105	120		116	89	100	100	99	109	116	90	94
10.2.0.57	0	0	9	30	1	1	6		93	96	95	93	98	104	116	95
10.2.0.59	36	21	25	23	24	20	22	21		145	132	144	141	139	126	128
10.2.0.61	9	15	12	5	1	0	4	8	59		45	63	54	60	59	59
10.2.0.63	16	15	15	10	9	10	15	14	107	103		117	117	114	116	120
10.2.0.65	4	0	4	7	7	6	3	4	80	83	82		79	88	85	82
10.2.0.67	5	8	8	5	3	14	9	9	93	97	113	89		96	92	87
10.2.0.69	5	10	8	10	17	13	12	12	118	110	111	107	127		125	114
10.2.0.71	8	10	9	12	12	13	7	13	140	137	140	151	135	144		155
10.2.0.73	4	5	3	1	2	2	3	5	121	97	119	109	108	112	124	

図 16　伝搬遅延が 1ms の場合 TCP Westwood の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	22	44	8	0	23	107	22	0	511	431	215	201	0	168	376	36
10.2.0.43		126	135	139	138	135	134	123	0	0	0	0	0	2	1	0
10.2.0.45	148		165	171	157	166	160	159	2	1	0	9	0	0	0	0
10.2.0.47	103	119		111	117	107	115	116	1	0	0	0	0	1	0	0
10.2.0.49	113	118	113		107	116	104	124	0	0	1	1	1	1	8	0
10.2.0.51	98	98	111	108		109	105	107	0	0	0	0	2	0	0	0
10.2.0.53	92	112	103	105	105		102	101	0	1	0	8	8	0	0	0
10.2.0.55	118	125	122	136	133	130		117	8	0	0	0	1	0	0	0
10.2.0.57	129	122	126	123	110	107	100		0	0	3	1	0	1	2	1
10.2.0.59	79	81	99	65	67	75	76	92		369	415	383	314	432	390	368
10.2.0.61	87	89	103	77	101	104	69	92	185		224	241	269	234	234	182
10.2.0.63	80	78	99	83	87	102	113	100	294	214		228	206	237	187	279
10.2.0.65	98	111	91	85	89	95	108	98	192	130	180		178	172	162	234
10.2.0.67	87	102	84	86	92	94	101	100	15	12	28	4		20	25	24
10.2.0.69	113	101	86	94	70	80	93	91	7	54	55	48	55		55	62
10.2.0.71	114	92	91	105	82	93	82	96	344	340	335	346	375	374		381
10.2.0.73	107	93	93	106	107	100	98	103	9	15	114	105	168	94	137	

図 17　伝搬遅延が 1ms の場合 TCP Vegas の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	44	26	111	28	23	0	39	0	332	603	406	4	59	286	0	0
10.2.0.43		126	92	129	128	106	122	108	12	4	10	5	17	10	16	16
10.2.0.45	119		135	141	132	124	121	135	13	20	13	19	12	19	14	13
10.2.0.47	96	129		111	121	107	116	114	1	2	1	5	4	4	1	2
10.2.0.49	110	121	120		117	119	103	126	3	9	4	7	7	7	8	8
10.2.0.51	98	98	105	105		108	97	103	5	8	6	5	6	2	7	4
10.2.0.53	81	76	88	73	76		86	73	3	1	2	2	1	2	2	5
10.2.0.55	133	138	141	142	131	139		133	15	16	15	13	12	13	12	9
10.2.0.57	77	81	78	83	82	81	81		0	0	0	0	0	0	0	0
10.2.0.59	100	104	95	110	87	89	99	90		320	327	342	302	281	288	0
10.2.0.61	108	120	91	131	118	117	126	112	479		513	436	418	381	310	471
10.2.0.63	106	105	93	110	113	113	105	107	389	328		403	388	343	381	548
10.2.0.65	100	99	96	102	101	105	104	103	1	3	0		31	45	11	110
10.2.0.67	100	88	128	117	110	102	125	107	67	55	58	69		65	113	82
10.2.0.69	107	91	119	91	96	95	111	116	250	216	213	273	260		279	289
10.2.0.71	105	103	104	103	111	105	101	100	3	5	5	2	4	1		25
10.2.0.73	110	111	107	101	100	99	116	93	3	0	4	3	61	123	146	

図 18　伝搬遅延が 1ms の場合 TCP Illinois の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	0	64	22	0	0	19	28	16	353	365	0	506	463	0	11	0
10.2.0.43	131	133	152	131	142	131	85	0	2	1	1	2	7	6	3	
10.2.0.45	144	106	122	112	124	127	110	25	17	29	26	28	17	31	31	
10.2.0.47	114	120	115	120	119	129	116	0	6	13	13	3	9	5	7	
10.2.0.49	99	84	83	90	86	95	94	1	7	2	9	2	4	2	1	
10.2.0.51	123	109	114	114	102	115	119	0	1	1	0	0	2	1	2	
10.2.0.53	156	148	146	132	140	150	154	12	14	4	3	10	3	9	14	
10.2.0.55	127	106	116	113	108	111	127	17	3	13	5	14	13	3	9	
10.2.0.57	65	95	88	102	95	94	99	1	10	4	7	1	0	2	0	
10.2.0.59	77	93	93	97	95	108	107	105	331	384	332	251	378	338	280	
10.2.0.61	89	106	91	96	120	94	0	101	215	176	234	268	137	119	133	
10.2.0.63	89	89	108	112	84	83	99	102	83	5	53	73	94	99	133	
10.2.0.65	99	93	96	95	86	100	89	106	470	423	502	447	481	506	457	
10.2.0.67	96	94	91	95	93	103	107	103	380	399	443	392	420	440	415	
10.2.0.69	112	88	110	91	111	101	106	94	5	2	8	1	6	2	36	
10.2.0.71	104	95	102	93	99	124	101	86	8	0	3	1	5	9	61	
10.2.0.73	100	93	94	91	104	105	89	94	0	1	1	1	2	0	0	

図 19　伝搬遅延が 5ms の場合 TCP New Reno の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	62	34	26	0	16	119	22	27	750	0	0	335	39	548	0	0
10.2.0.43	114	104	89	93	87	100	100	18	15	28	29	32	23	13	22	
10.2.0.45	84	84	75	82	82	87	79	22	18	17	13	18	15	8	14	
10.2.0.47	102	86	88	92	94	83	82	17	14	12	16	8	12	11	10	
10.2.0.49	75	82	76	83	77	87	81	3	2	0	2	0	0	0	6	
10.2.0.51	64	74	67	66	74	70	66	2	2	3	6	2	1	8	6	
10.2.0.53	82	86	88	76	90	88	90	2	0	1	2	0	0	6	1	
10.2.0.55	92	99	91	101	100	85	103	6	7	5	5	5	3	8	6	
10.2.0.57	98	78	81	94	85	84	87	10	15	12	15	12	10	14	10	
10.2.0.59	122	127	116	124	109	138	124	118	696	693	682	649	563	549	636	
10.2.0.61	114	135	125	141	129	124	140	114	7	3	7	49	117	168	111	
10.2.0.63	125	124	126	123	124	150	125	122	0	0	3	5	12	19	11	
10.2.0.65	121	108	131	131	116	138	126	116	273	292	294	281	260	260	294	
10.2.0.67	133	119	122	118	125	99	118	115	13	21	17	7	13	23	17	
10.2.0.69	115	105	127	113	106	106	112	127	460	496	494	466	475	472	438	
10.2.0.71	96	94	94	104	98	96	100	119	0	0	0	2	0	0	2	
10.2.0.73	106	103	101	123	119	130	104	109	1	0	0	0	0	0	27	

図 20　伝搬遅延が 5ms の場合 BIC-TCP の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	522	161	497	438	0	70	0	0	29	80	5	0	11	17	21	22
10.2.0.43	414	402	415	166	157	201	388	124	110	76	77	84	92	83	81	
10.2.0.45	155	145	154	34	107	48	129	98	116	101	98	89	92	72	110	
10.2.0.47	410	428	408	423	521	497	343	98	109	105	99	95	95	82	104	
10.2.0.49	361	394	377	620	505	453	338	107	102	105	98	102	95	85	122	
10.2.0.51	0	22	20	21	1	27	110	101	76	91	94	103	100	82	100	
10.2.0.53	28	50	29	33	112	76	140	93	91	102	80	101	90	86	89	
10.2.0.55	0	0	2	0	3	7	11	103	91	90	92	88	100	102	88	
10.2.0.57	0	1	0	0	102	96	152	92	88	90	84	92	109	114	94	
10.2.0.59	27	13	26	21	17	12	18	21	148	132	143	140	136	164	149	
10.2.0.61	41	38	34	44	33	18	33	29	170	151	155	156	129	166	165	
10.2.0.63	3	16	14	9	5	14	13	7	93	97	97	97	102	94	91	
10.2.0.65	1	17	13	9	16	20	8	8	58	55	58	60	74	69	56	
10.2.0.67	0	3	1	4	13	14	10	9	88	86	86	82	89	89	87	
10.2.0.69	16	9	7	9	8	11	16	19	147	146	156	160	149	153	133	
10.2.0.71	0	0	1	1	0	8	12	16	76	89	100	99	89	102	98	
10.2.0.73	14	16	16	9	26	26	16	15	112	116	118	122	127	135	135	

図 21　伝搬遅延が 5ms の場合 TCP Westwood の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	50	0	0	22	39	11	17	99	387	155	212	530	0	426	0	0
10.2.0.43		147	158	150	151	131	151	143	5	2	3	4	10	2	4	0
10.2.0.45	99		85	98	94	74	91	93	1	4	0	2	5	1	1	0
10.2.0.47	152	137		147	127	150	153	162	3	9	0	0	0	0	0	6
10.2.0.49	124	135	132		116	115	117	128	3	2	1	2	1	1	13	5
10.2.0.51	97	129	103	106		110	112	127	0	0	13	9	6	4	7	2
10.2.0.53	150	154	144	139	148		123	171	4	4	0	1	0	6	4	0
10.2.0.55	103	103	113	90	112	110		111	5	2	6	1	1	3	0	3
10.2.0.57	139	128	158	155	142	146	136		1	7	0	0	1	3	1	8
10.2.0.59	89	108	72	127	101	103	92	69		341	373	353	373	381	284	193
10.2.0.61	100	85	70	90	89	73	101	55	154		110	154	114	138	159	123
10.2.0.63	68	88	84	101	100	101	89	85	143	139		145	163	156	176	146
10.2.0.65	98	107	76	60	40	105	80	88	417	410	408		402	372	399	470
10.2.0.67	94	101	103	108	112	108	104	92	28	17	0	10		9	40	63
10.2.0.69	93	103	97	92	95	101	100	107	398	411	377	349	419		465	505
10.2.0.71	109	93	106	91	93	106	94	88	26	67	59	10	69	70		57
10.2.0.73	102	92	110	90	100	109	99	107	3	7	6	4	14	12	38	

図 22　伝搬遅延が 5ms の場合 TCP Vegas の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	17	0	60	24	35	0	22	0	526	0	0	541	155	156	323	0
10.2.0.43		99	94	87	95	88	98	88	0	2	3	0	3	16	10	14
10.2.0.45	116		112	109	114	109	125	114	5	5	2	6	16	4	5	6
10.2.0.47	136	145		113	131	126	140	122	26	25	26	23	16	21	21	20
10.2.0.49	66	67	67		60	63	61	60	0	2	1	4	1	8	6	9
10.2.0.51	144	145	145	143		154	159	145	14	15	17	15	20	12	9	3
10.2.0.53	131	127	130	138	128		133	139	3	2	2	1	1	0	0	5
10.2.0.55	145	142	151	154	166	162		173	4	9	9	4	7	0	0	1
10.2.0.57	73	72	78	90	76	82	73		4	0	2	2	3	0	0	0
10.2.0.59	85	93	92	83	89	90	100	91		463	450	433	430	475	441	364
10.2.0.61	84	103	97	79	107	99	111	86	22		10	10	59	48	74	70
10.2.0.63	98	84	95	88	100	85	90	104	3	0		4	7	4	11	16
10.2.0.65	96	102	84	77	98	95	103	84	448	485	485		456	429	379	496
10.2.0.67	77	96	93	78	88	103	85	90	123	146	145	145		132	121	149
10.2.0.69	101	100	82	100	92	102	93	91	141	147	154	146	144		142	135
10.2.0.71	91	99	87	98	103	110	88	94	254	276	273	235	253	263		287
10.2.0.73	105	94	96	104	88	99	85	84	2	0	0	9	14	6	30	

図 23　伝搬遅延が 5ms の場合 TCP Illinois の各ピアのピース取得状況

4.1.2 ピア平均スループット測定の結果

図 24 は伝搬遅延が異なる場合 TCP 輻輳制御アルゴリズム毎の再送回数を示す。図 25 と図 26 は伝搬遅延が 1ms と 5ms の場合のスループット累積分布関数を示す。図 27 と図 28 は伝搬遅延が 1ms と 5ms の場合の TCP フロー毎の TCP フロー継続時間の累積分布関数を示す。図 29 と図 30 は伝搬遅延が 1ms と 5ms の場合の TCP フロー毎の TCP フロースループットの累積分布関数を示す。伝搬遅延が 5ms の場合、BIC-TCP のスループットが遅くなる。伝搬遅延が 5ms の場合、図 24 に示されたように、BIC-TCP の再送回数が多くなるため、BIC-TCP の TCP フロー継続時間が長くなる。また、前節で説明したように、伝搬遅延が 5ms の場合、BIC-TCP のローカライゼーション度合いも悪くなるため、ピア平均スループットが遅くなる。

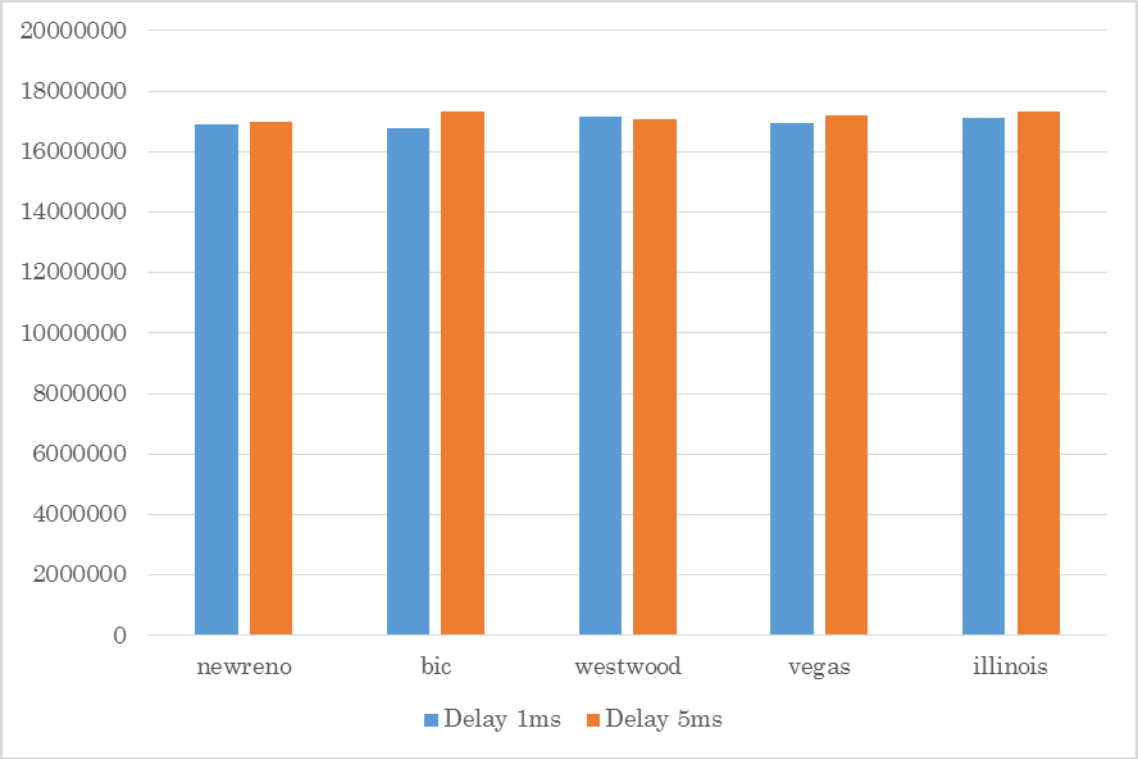

図 24　伝搬遅延が異なる場合 TCP 輻輳制御アルゴリズム毎の再送回数

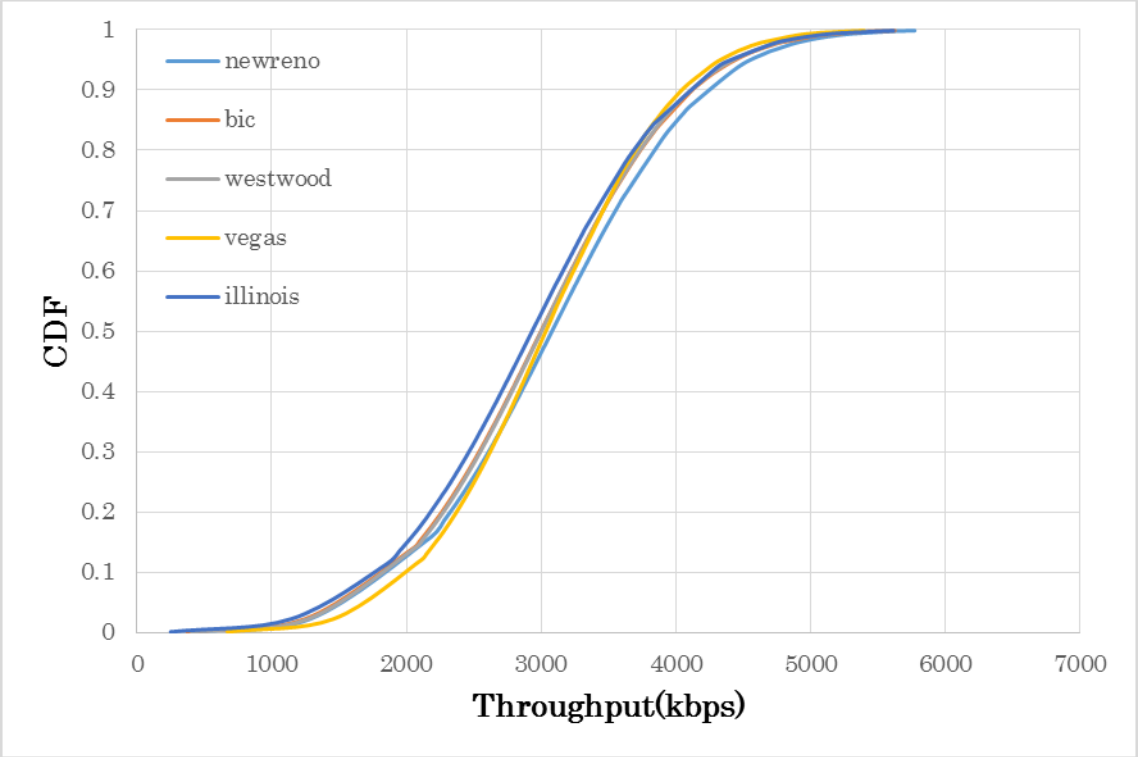

図 25　伝搬遅延が 1ms の場合ピア平均スループット累積分布関数

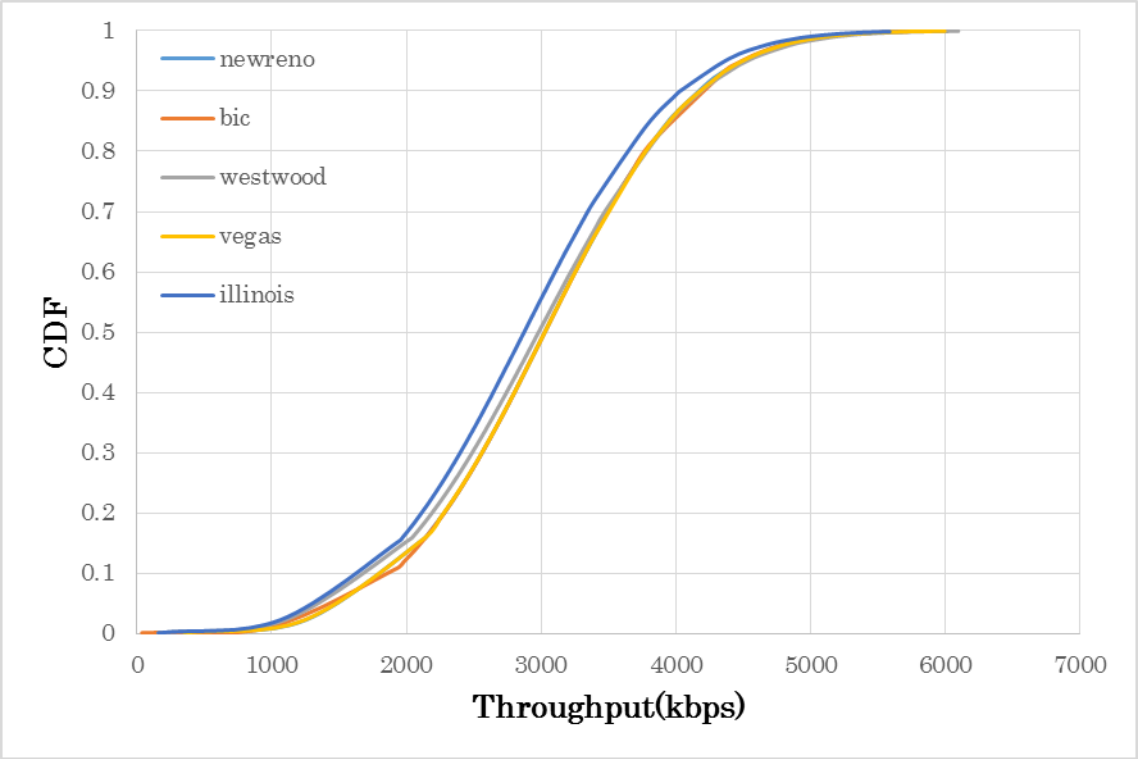

図 26　伝搬遅延が 5ms の場合ピア平均スループット累積分布関数

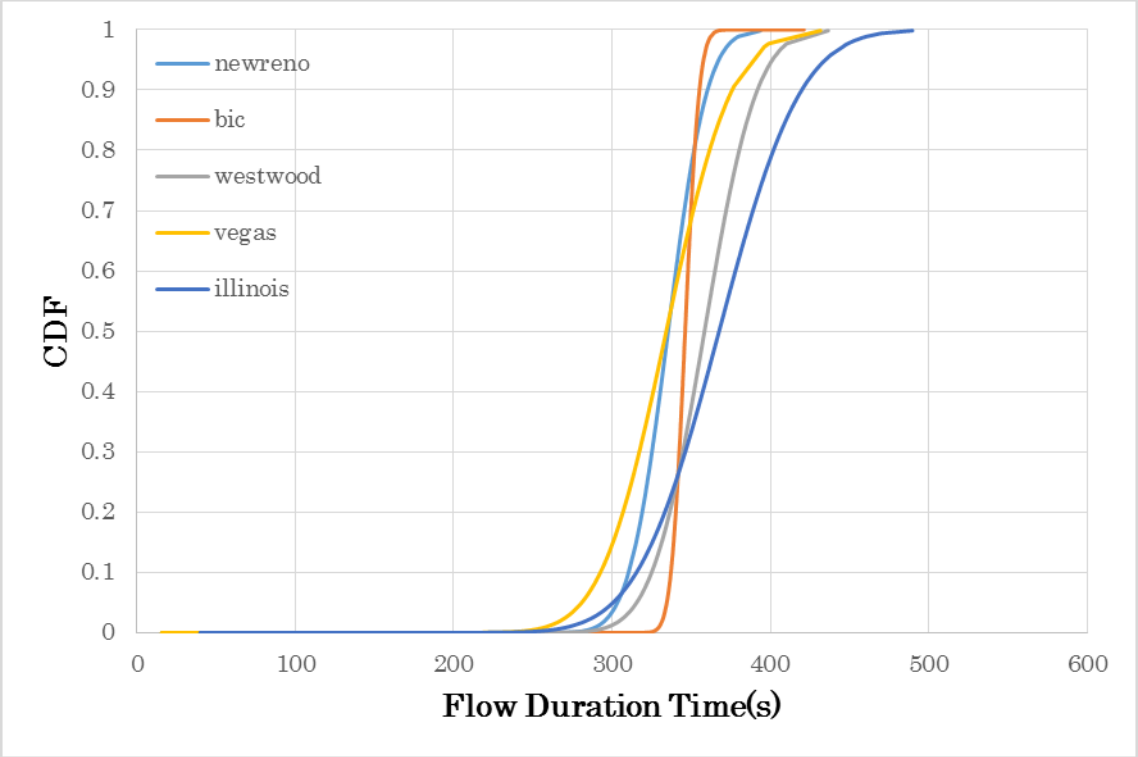

図 27　伝搬遅延が 1ms の場合 TCP フロー毎のフロー継続時間累積分布関数

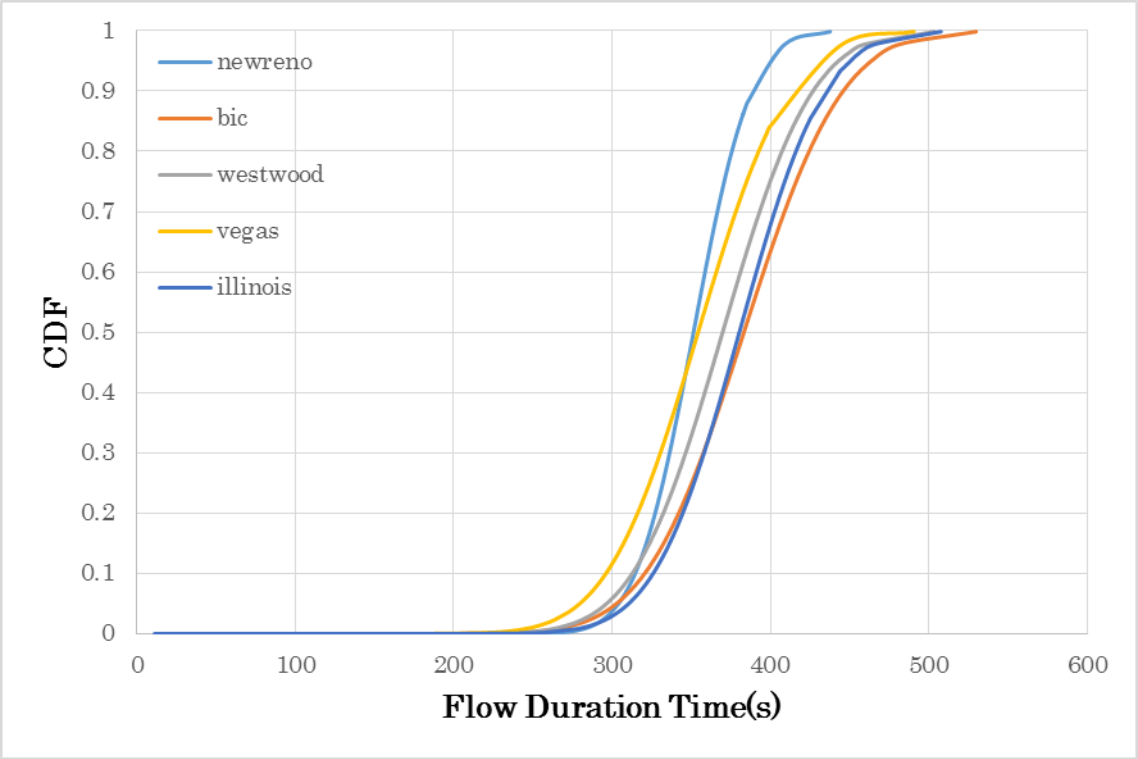

図 28　伝搬遅延が 5ms の場合 TCP フロー毎のフロー継続時間累積分布関数

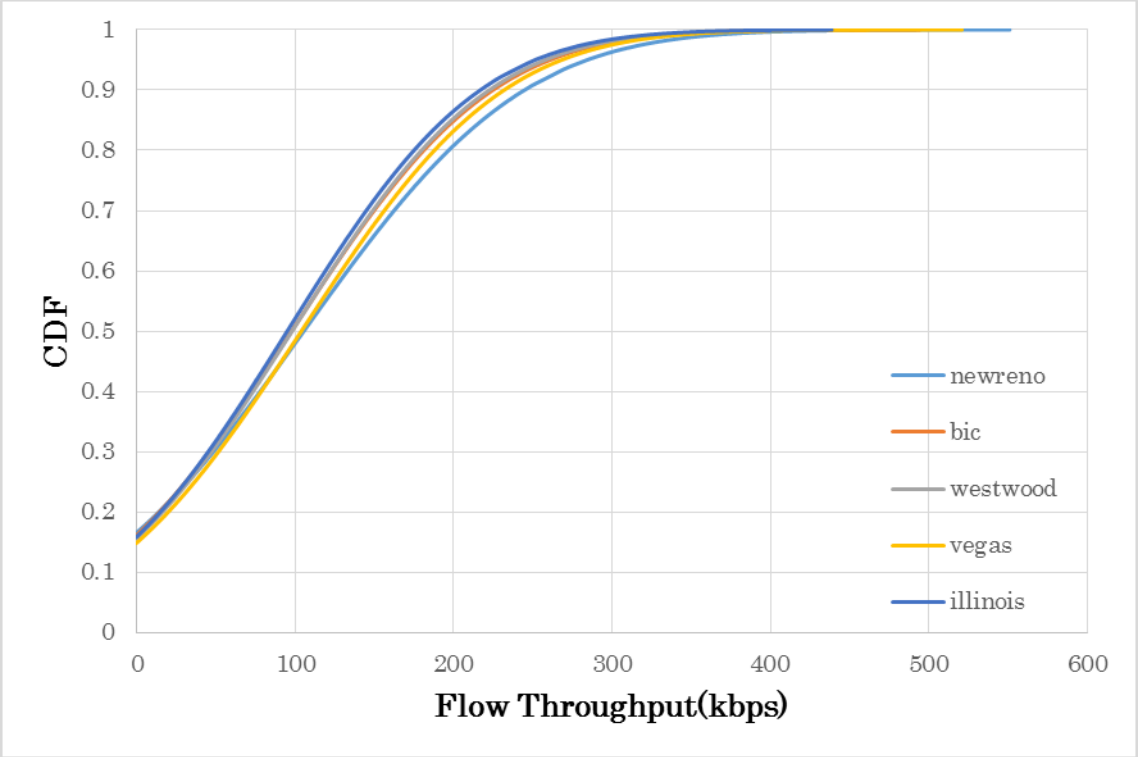

図 29　伝搬遅延が 1ms の場合 TCP フロー毎のフロースループット累積分布関数

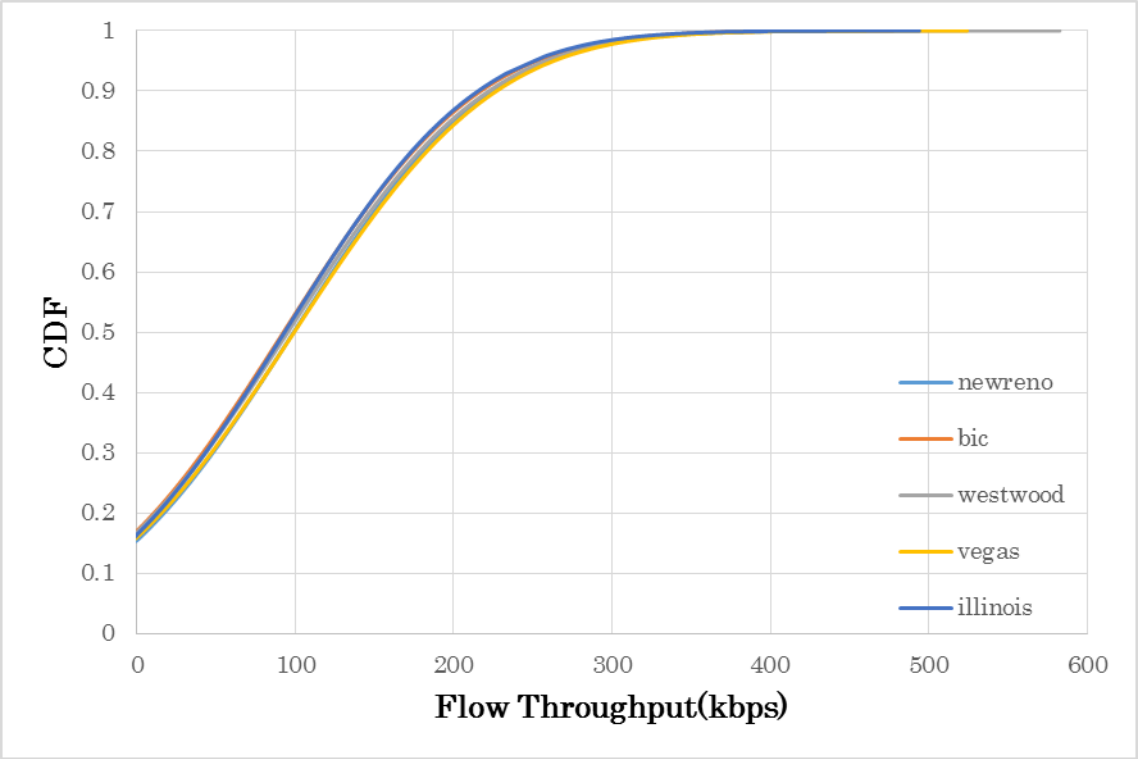

図 30　伝搬遅延が 5ms の場合 TCP フロー毎のフロースループット累積分布関数

4.2 AS 内の帯域幅が異なる場合

本実験では、AS 内の帯域幅が異なる場合、TCP バージョンごとにそのスループットとローカライゼーション度合いを比較する。3 章で述べた基本トポロジーの AS 内の帯域幅を 100Mbps と 10Mbps に変更し、その結果を考察する。

4.2.1 ローカライゼーション度合い測定の結果

図 31 と図 32 は帯域幅が 100Mbps と 10Mbps の場合のローカライゼーション度合いの累積分布関数を示す。図 33 と図 34 は帯域幅が 100Mbps と 10Mbps の場合のダウンロードしたピースの平均ホップ数の累積分布関数を示す。帯域幅が 100Mbps の場合 TCP Vegas がローカライズされやすい。図 35 から図 44 に示された各ピアのピース取得状況のマトリックスを見ると、最初にシーダから非チョークされてないピアは、AS 外から持ってきたピースを他の TCP 輻輳制御アルゴリズムより AS 内で積極的にダウンロードする。それで、TCP Vegas のダウンロードしたピース平均ホップ数も少ない。帯域幅が 10Mbps の場合 TCP Westwood の性能が悪い。TCP Westwood の場合、帯域幅が 10Mbps になると、シーダから選ばれたピアの AS において、ピース数が多いため、AS 内から積極的にピースをダウンロードするが、ピースが少ない AS のピアは AS 内と AS 外の帯域幅条件が同一になるため、帯域幅が 100Mbps の場合より、AS 外からのダ

ウンロードが増える。そのため、ローカライゼーション度合いが悪くなり、ダウンロードしたピースの平均ホップ数も高くなる。

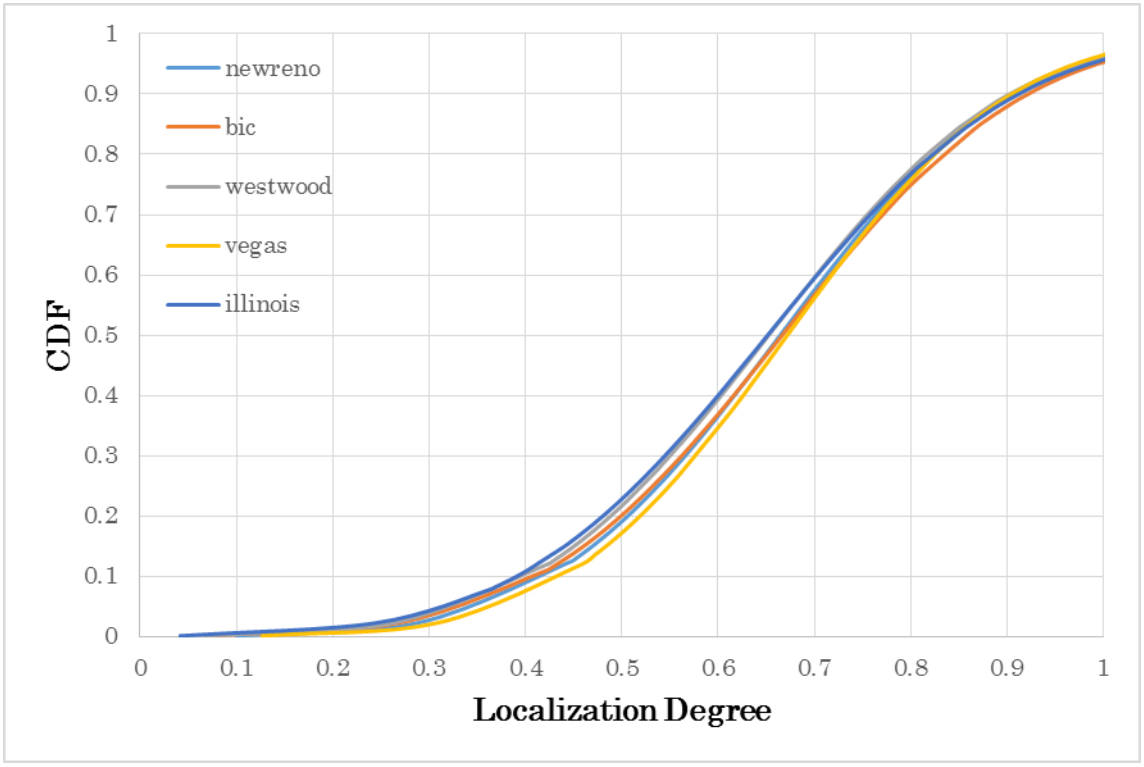

図 31　帯域幅が 100Mbps の場合ローカライゼーション度合い累積分布関数

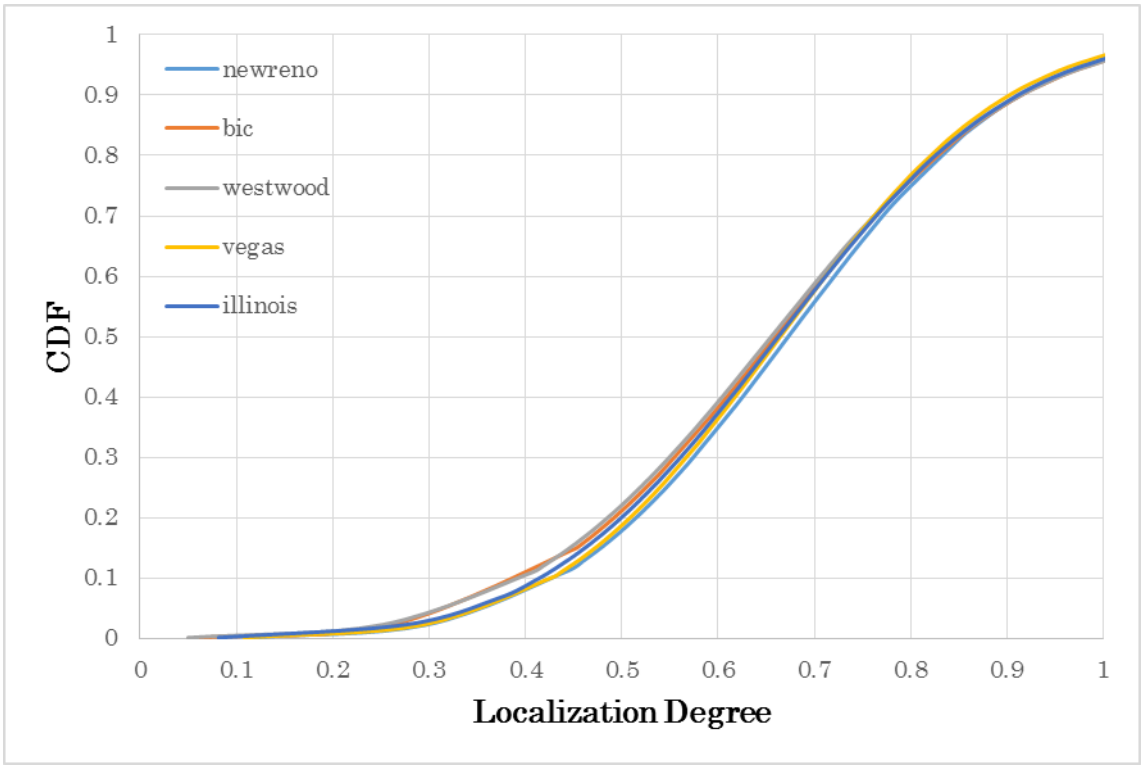

図 32　帯域幅が 10Mbps の場合ローカライゼーション度合い累積分布関数

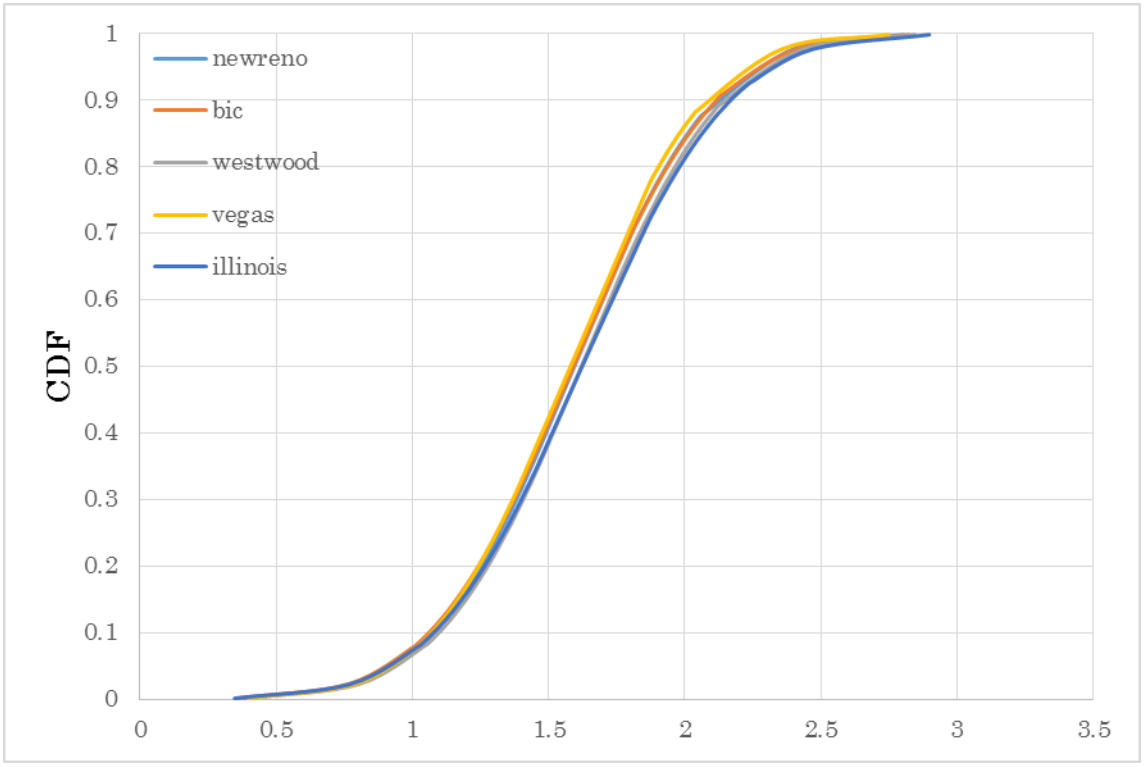

図 33　帯域幅が 100Mbps の場合ダウンロードしたピースの平均ホップ数累積分布関数

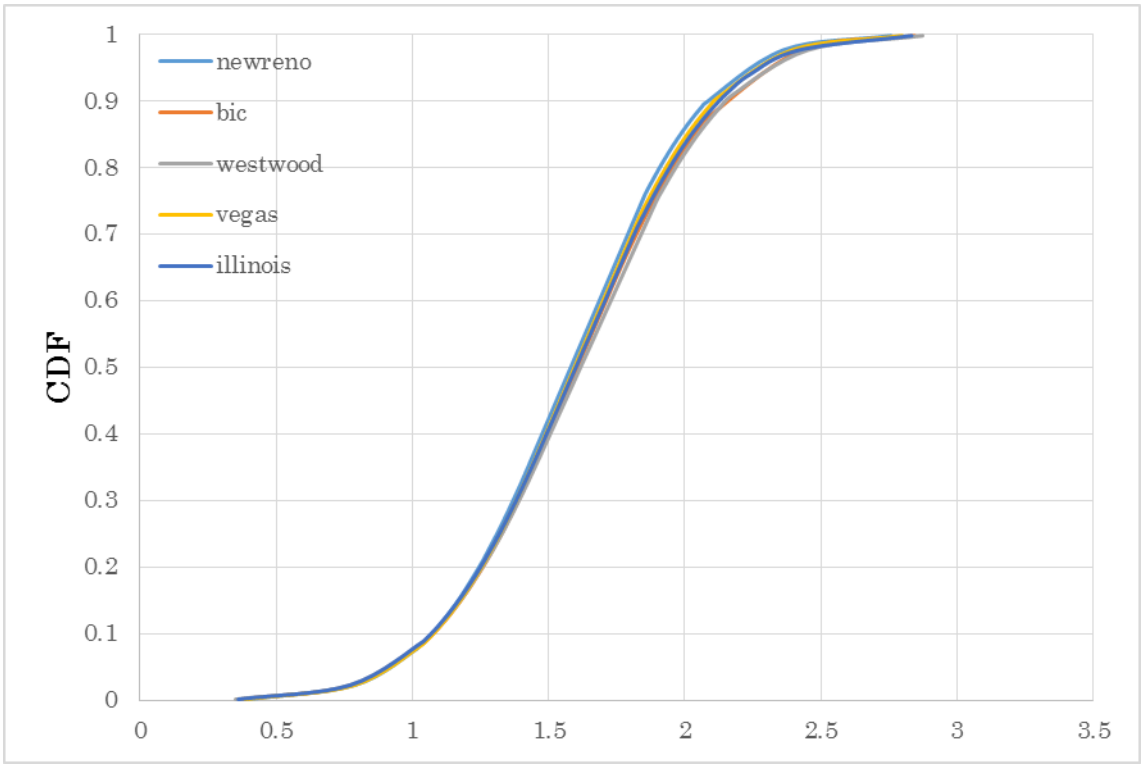

図 34　帯域幅が 10Mbps の場合ダウンロードしたピースの平均ホップ数累積分布関数

図 35 から図 44 は帯域幅が 100Mbps と 10Mbps の場合、TCP 輻輳制御アルゴリズムごとの各ピアのピース取得状況を示している。縦軸は送信側のピアの IP アドレスであり、横軸は受信側

のピアの IP アドレスである。横軸の IP アドレスの”10.2.0”は省略する。IP アドレスが”10.2.0.41”のピアはシーダである。表の数値は各ピアからダウンロードしたピース数を示している。紫色の部分は AS 内ピアからダウンロードしたピース数を表し、緑色は AS 外からの取得ピース数を表している。

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	15	45	0	35	25	1	30	0	333	611	0	492	0	209	0	248
10.2.0.43		106	116	109	108	122	116	113	6	0	3	5	1	1	2	4
10.2.0.45	114		120	113	101	113	123	99	18	18	18	20	16	19	19	16
10.2.0.47	112	110		106	102	110	115	112	2	0	4	4	4	0	2	3
10.2.0.49	125	126	139		128	130	116	129	12	14	19	8	14	16	10	13
10.2.0.51	134	111	140	120		135	132	136	7	5	9	7	9	8	9	7
10.2.0.53	120	108	131	122	118		120	118	2	0	1	1	3	1	1	0
10.2.0.55	82	94	93	88	89	100		99	14	16	12	15	16	15	13	15
10.2.0.57	82	77	86	78	91	79	76		4	0	2	0	1	0	0	0
10.2.0.59	118	126	71	96	88	97	98	112		312	297	315	252	303	215	217
10.2.0.61	101	111	98	109	95	93	104	108	561		578	540	498	561	609	436
10.2.0.63	91	101	100	100	106	97	107	104	0	8		0	34	3	42	42
10.2.0.65	95	104	102	103	106	91	96	94	427	409	434		456	430	440	450
10.2.0.67	98	94	99	95	119	112	101	95	1	1	1	0		1	1	2
10.2.0.69	112	95	90	96	105	101	91	94	4	2	2	0	4		4	19
10.2.0.71	114	100	101	102	96	104	94	99	5	0	2	2	65	10		110
10.2.0.73	106	94	96	96	98	104	93	93	193	179	204	171	209	208	206	

図 35　帯域幅が 100Mbps の場合 TCP New Reno の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	553	481	55	102	0	2	4	516	0	21	0	36	171	0	20	39
10.2.0.43		470	431	390	279	198	191	488	98	91	59	70	68	66	103	64
10.2.0.45	394		412	442	370	253	366	413	93	148	91	70	41	90	89	67
10.2.0.47	44	44		0	71	164	136	0	111	113	124	98	78	85	91	72
10.2.0.49	92	97	109		188	240	256	98	94	102	102	122	95	80	81	127
10.2.0.51	0	0	4	57		27	28	2	105	110	86	110	105	85	91	123
10.2.0.53	0	0	2	15	2		4	14	83	110	117	103	119	98	100	111
10.2.0.55	0	0	11	17	15	28		10	99	93	119	100	103	130	85	114
10.2.0.57	452	446	510	517	617	629	558		101	98	115	94	98	116	117	91
10.2.0.59	4	6	1	0	4	2	0	4		61	78	77	82	78	66	62
10.2.0.61	3	2	4	9	8	2	6	6	141		131	142	160	136	142	144
10.2.0.63	2	2	13	2	0	7	9	7	115	118		95	123	114	121	119
10.2.0.65	14	1	9	14	12	9	3	0	116	102	109		117	134	128	151
10.2.0.67	5	0	2	6	4	5	1	8	110	109	109	101		128	119	105
10.2.0.69	9	2	0	1	3	0	0	8	90	86	94	94	90		83	93
10.2.0.71	8	7	9	0	1	9	8	12	127	122	121	126	164	141		136
10.2.0.73	0	22	8	1	13	4	8	2	149	174	151	171	154	146	169	

図 36　帯域幅が 100Mbps の場合 BIC-TCP の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	546	532	0	0	447	0	158	3	71	0	20	0	23	24	32	16
10.2.0.43		499	468	262	510	524	247	324	98	108	116	75	108	80	95	110
10.2.0.45	462		424	413	436	472	533	548	104	108	102	82	89	82	97	98
10.2.0.47	0	0		18	0	24	31	23	105	103	104	100	83	90	96	94
10.2.0.49	2	0	1		0	1	0	1	96	97	104	118	88	92	108	100
10.2.0.51	381	377	403	429		358	388	386	101	96	86	111	92	114	102	99
10.2.0.53	2	0	51	148	2		141	105	92	87	99	110	117	102	95	115
10.2.0.55	107	90	145	204	105	120		116	89	100	100	99	109	116	90	94
10.2.0.57	0	0	9	30	1	1	6		93	96	95	93	98	104	116	95
10.2.0.59	36	21	25	23	24	20	22	21		145	132	144	141	139	126	128
10.2.0.61	9	15	12	5	1	0	4	8	59		45	63	54	60	59	59
10.2.0.63	16	15	15	10	9	10	15	14	107	103		117	117	114	116	120
10.2.0.65	4	0	4	7	7	6	3	4	80	83	82		79	88	85	82
10.2.0.67	5	8	8	5	3	14	9	9	93	97	113	89		96	92	87
10.2.0.69	5	10	8	10	17	13	12	12	118	110	111	107	127		125	114
10.2.0.71	8	10	9	12	12	13	7	13	140	137	140	151	135	144		155
10.2.0.73	4	5	3	1	2	2	3	5	121	97	119	109	108	112	124	

図 37　帯域幅が 100Mbps の場合 TCP Westwood の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	22	44	8	0	23	107	22	0	511	431	215	201	0	168	376	36
10.2.0.43		126	135	139	138	135	134	123	0	0	0	0	0	2	1	0
10.2.0.45	148		165	171	157	166	160	159	2	1	0	9	0	0	0	0
10.2.0.47	103	119		111	117	107	115	116	1	0	0	0	0	1	0	0
10.2.0.49	113	118	113		107	116	104	124	0	0	1	1	1	1	8	0
10.2.0.51	98	98	111	108		109	105	107	0	0	0	0	2	0	0	0
10.2.0.53	92	112	103	105	105		102	101	0	1	0	8	8	0	0	0
10.2.0.55	118	125	122	136	133	130		117	8	0	0	0	1	0	0	0
10.2.0.57	129	122	126	123	110	107	100		0	0	3	1	0	1	2	1
10.2.0.59	79	81	99	65	67	75	76	92		369	415	383	314	432	390	368
10.2.0.61	87	89	103	77	101	104	69	92	185		224	241	269	234	234	182
10.2.0.63	80	78	99	83	87	102	113	100	294	214		228	206	237	187	279
10.2.0.65	98	111	91	85	89	95	108	98	192	130	180		178	172	162	234
10.2.0.67	87	102	84	86	92	94	101	100	15	12	28	4		20	25	24
10.2.0.69	113	101	86	94	70	80	93	91	7	54	55	48	55		55	62
10.2.0.71	114	92	91	105	82	93	82	96	344	340	335	346	375	374		381
10.2.0.73	107	93	93	106	107	100	98	103	9	15	114	105	168	94	137	

図 38　帯域幅が 100Mbps の場合 TCP Vegas の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	44	26	111	28	23	0	39	0	332	603	406	4	59	286	0	0
10.2.0.43		126	92	129	128	106	122	108	12	4	10	5	17	10	16	16
10.2.0.45	119		135	141	132	124	121	135	13	20	13	19	12	19	14	13
10.2.0.47	96	129		111	121	107	116	114	1	2	1	5	4	4	1	2
10.2.0.49	110	121	120		117	119	103	126	3	9	4	7	7	7	8	8
10.2.0.51	98	98	105	105		108	97	103	5	8	6	5	6	2	7	4
10.2.0.53	81	76	88	73	76		86	73	3	1	2	2	1	2	2	5
10.2.0.55	133	138	141	142	131	139		133	15	16	15	13	12	13	12	9
10.2.0.57	77	81	78	83	82	81	81		0	0	0	0	0	0	0	0
10.2.0.59	100	104	95	110	87	89	99	90		320	327	342	302	281	288	0
10.2.0.61	108	120	91	131	118	117	126	112	479		513	436	418	381	310	471
10.2.0.63	106	105	93	110	113	113	105	107	389	328		403	388	343	381	548
10.2.0.65	100	99	96	102	101	105	104	103	1	3	0		31	45	11	110
10.2.0.67	100	88	128	117	110	102	125	107	67	55	58	69		65	113	82
10.2.0.69	107	91	119	91	96	95	111	116	250	216	213	273	260		279	289
10.2.0.71	105	103	104	103	111	105	101	100	3	5	5	2	4	1		25
10.2.0.73	110	111	107	101	100	99	116	93	3	0	4	3	61	123	146	

図 39　帯域幅が 100Mbps の場合 TCP Illinois の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	530	553	286	0	336	0	0	13	42	30	2	27	0	34	2	78
10.2.0.43		368	381	373	411	439	416	345	82	78	98	107	65	74	89	66
10.2.0.45	383		407	497	416	429	481	295	89	78	101	101	80	75	83	83
10.2.0.47	169	211		275	262	244	251	286	104	120	101	99	81	78	99	93
10.2.0.49	32	37	31		6	0	8	46	94	101	141	96	99	128	110	96
10.2.0.51	259	264	325	324		327	308	314	81	95	90	93	91	88	87	99
10.2.0.53	51	50	36	38	33		24	74	102	118	109	115	102	117	113	106
10.2.0.55	64	26	43	11	36	52		137	109	111	107	119	104	102	101	110
10.2.0.57	42	22	21	8	27	5	12		113	99	108	95	112	96	98	107
10.2.0.59	10	12	14	12	17	19	9	18		109	123	113	132	122	125	121
10.2.0.61	9	9	19	13	14	9	8	17	97		89	94	129	102	101	100
10.2.0.63	7	3	0	1	0	6	3	4	108	118		111	126	111	112	107
10.2.0.65	6	2	3	8	5	9	9	7	130	126	126		139	124	134	139
10.2.0.67	1	4	0	0	0	0	10	5	110	107	109	113		124	110	134
10.2.0.69	16	17	9	11	12	21	17	13	113	105	106	114	113		112	103
10.2.0.71	1	3	2	2	0	6	5	7	105	108	107	99	109	104		105
10.2.0.73	0	1	2	1	0	7	13	6	87	83	78	80	100	88	78	

図 40　帯域幅が 10Mbps の場合 TCP New Reno の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	56	24	22	25	0	0	0	0	573	0	17	480	561	0	94	0
10.2.0.43		115	112	109	111	103	120	104	32	26	7	13	22	6	8	21
10.2.0.45	117		132	127	127	138	118	119	8	9	0	4	4	8	1	4
10.2.0.47	117	116		98	114	111	115	98	6	5	5	5	1	6	8	5
10.2.0.49	81	86	87		91	87	98	81	3	9	16	9	8	12	15	10
10.2.0.51	98	110	82	90		94	93	98	0	0	6	3	2	3	8	3
10.2.0.53	128	142	129	129	132		127	139	1	0	2	6	2	4	6	2
10.2.0.55	155	132	121	102	120	123		113	1	1	8	6	8	7	2	2
10.2.0.57	83	93	84	93	91	87	86		0	1	0	0	0	0	0	2
10.2.0.59	83	88	79	81	85	83	82	83		529	526	394	349	527	499	517
10.2.0.61	132	122	127	119	104	109	117	124	81		16	70	66	19	13	8
10.2.0.63	90	98	99	109	129	119	117	112	72	38		72	58	37	8	35
10.2.0.65	82	85	95	77	82	90	95	86	292	438	436		339	451	436	439
10.2.0.67	72	105	74	89	70	70	80	82	335	400	447	357		397	441	450
10.2.0.69	85	107	114	110	116	118	126	119	57	11	6	33	49		18	9
10.2.0.71	79	98	106	105	103	115	107	103	76	79	62	90	84	76		72
10.2.0.73	111	110	112	113	113	126	120	125	41	26	22	39	30	25	26	

図 41　帯域幅が 10Mbps の場合 BIC-TCP の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	58	19	17	5	102	38	22	15	0	0	0	82	561	577	397	67
10.2.0.43		141	141	110	114	124	125	101	29	9	23	21	32	21	7	8
10.2.0.45	85		87	101	89	102	89	99	10	15	6	8	12	8	4	6
10.2.0.47	126	121		121	116	115	115	109	12	16	10	17	12	8	17	15
10.2.0.49	92	86	79		80	85	83	83	1	0	1	5	5	0	2	1
10.2.0.51	98	99	90	85		87	87	95	0	0	6	0	0	3	2	3
10.2.0.53	80	84	83	85	97		84	81	7	13	15	13	4	10	9	13
10.2.0.55	107	115	104	106	112	115		106	6	13	7	2	1	9	23	15
10.2.0.57	96	96	92	94	92	104	94		1	0	0	0	0	0	1	0
10.2.0.59	117	113	131	118	114	121	116	120		4	5	19	44	65	48	27
10.2.0.61	114	121	122	110	122	107	113	117	4		5	15	32	50	29	7
10.2.0.63	110	113	122	111	116	102	116	126	7	9		15	63	67	61	21
10.2.0.65	112	117	122	126	121	105	121	112	79	77	75		103	96	90	87
10.2.0.67	99	92	92	99	95	85	102	89	499	496	494	475		358	413	474
10.2.0.69	86	77	81	78	80	77	85	94	507	506	512	490	372		400	499
10.2.0.71	83	76	94	102	95	98	110	111	343	350	351	339	262	235		337
10.2.0.73	104	96	107	102	122	105	105	109	69	66	69	76	79	80	77	

図 42　帯域幅が 10Mbps の場合 TCP Westwood の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	0	17	10	24	0	37	38	25	519	0	342	527	0	151	0	187
10.2.0.43		119	119	113	118	114	98	136	1	9	0	2	0	2	1	0
10.2.0.45	114		109	100	91	99	108	98	3	1	0	2	0	2	1	2
10.2.0.47	95	93		87	99	95	105	104	0	3	1	1	0	1	3	4
10.2.0.49	120	122	111		94	123	127	127	9	1	2	2	12	0	0	3
10.2.0.51	128	130	125	124		122	123	118	1	4	2	2	7	6	1	0
10.2.0.53	121	128	118	121	135		117	130	8	7	6	3	9	7	13	4
10.2.0.55	120	114	115	112	115	133		124	5	8	14	5	6	6	7	5
10.2.0.57	153	151	136	125	137	114	127		5	10	5	24	7	9	7	18
10.2.0.59	73	86	76	89	98	134	112	116		462	378	334	448	400	265	380
10.2.0.61	128	106	109	99	114	108	112	111	76		86	85	14	41	0	40
10.2.0.63	91	91	94	95	83	89	100	90	278	329		233	330	317	331	316
10.2.0.65	85	79	78	78	90	82	100	89	360	471	351		473	426	469	420
10.2.0.67	111	106	110	114	108	104	105	104	48	0	50	32		30	86	21
10.2.0.69	100	93	101	110	102	105	106	109	142	169	167	140	178		252	131
10.2.0.71	98	102	114	101	122	115	124	108	0	0	67	66	0	60		47
10.2.0.73	65	109	126	118	122	106	85	86	127	109	105	123	94	119	136	

図 43　帯域幅が 10Mbps の場合 TCP Vegas の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	51	0	28	12	20	29	121	11	0	0	31	638	83	171	543	242
10.2.0.43		109	103	101	100	86	104	101	14	2	23	27	7	4	23	14
10.2.0.45	68		69	76	73	85	81	88	7	10	3	5	0	0	4	1
10.2.0.47	102	108		107	111	88	107	111	16	17	13	12	10	6	15	10
10.2.0.49	93	83	92		94	93	91	102	1	3	0	3	1	3	4	7
10.2.0.51	99	104	112	102		93	107	106	5	16	9	7	12	7	10	9
10.2.0.53	145	128	114	137	132		148	134	13	7	5	5	13	10	0	7
10.2.0.55	88	73	84	75	93	60		80	0	3	0	1	10	14	5	5
10.2.0.57	91	97	97	90	104	87	107		0	2	2	0	2	11	2	2
10.2.0.59	133	136	126	134	127	138	119	124		17	5	78	18	21	51	48
10.2.0.61	112	134	134	127	137	151	136	133	9		9	60	12	22	51	34
10.2.0.63	134	126	142	113	141	141	136	128	20	17		73	20	29	89	42
10.2.0.65	87	89	89	95	85	87	93	87	555	556	553		563	525	388	501
10.2.0.67	106	116	128	114	91	138	36	103	69	70	72	68		78	81	66
10.2.0.69	78	18	99	85	113	112	115	96	142	140	125	125	126		139	112
10.2.0.71	84	108	98	96	75	85	98	91	524	522	527	325	523	505		479
10.2.0.73	110	115	119	132	130	121	125	110	201	200	199	161	176	172	184	

図 44　帯域幅が 10Mbps の場合 TCP Illinois の各ピアのピース取得状況

4.2.2 ピア平均スループット測定の結果

図 45 は帯域幅が異なる場合 TCP 輻輳制御アルゴリズム毎の再送回数を示す。図 46 と図 47 は帯域幅が 100Mbps と 10Mbps の場合のスループット累積分布関数を示す。図 48 と図 49 は帯域幅が 100Mbps と 10Mbps の場合の TCP フロー毎の TCP フロー継続時間の累積分布関数を示す。図 50 と図 51 は帯域幅が 100Mbps と 10Mbps の場合の TCP フロー毎の TCP フロースループットの累積分布関数を示す。帯域幅が 10Mbps の場合、BIC-TCP のスループットがやや速くなる。帯域幅が 10Mbps の場合、BIC-TCP の再送回数がやや少なくなり、TCP フロー毎の継続時間も短くなり、帯域幅が 100Mbps の場合よりスループットが少し速くなる。

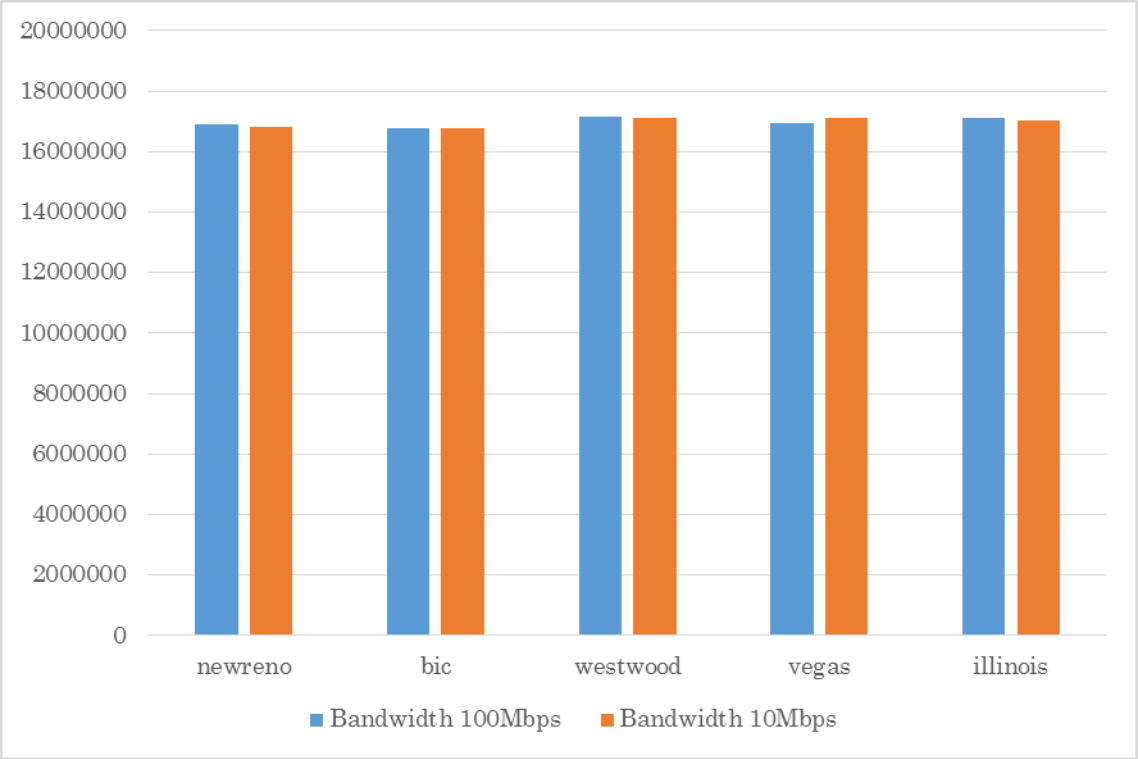

図 45　帯域幅が異なる場合 TCP 輻輳制御アルゴリズム毎の再送回数

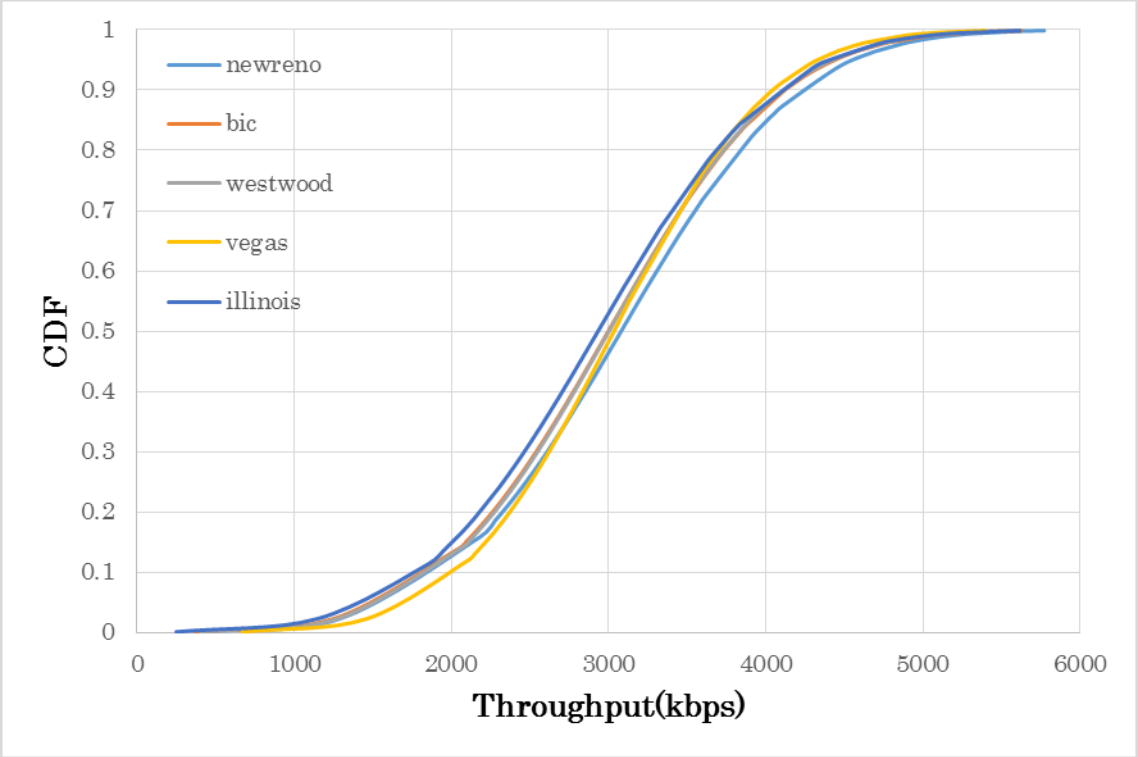

図 46　帯域幅が 100Mbps の場合ピア平均スループット累積分布関数

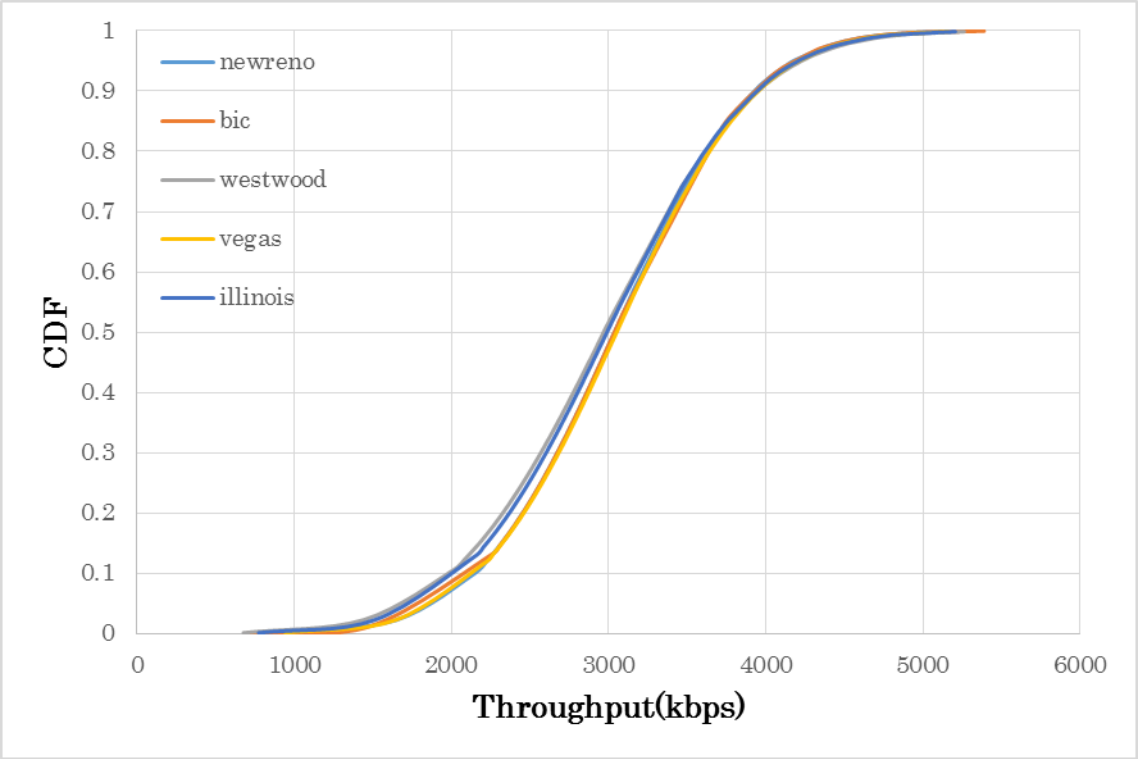

図 47　帯域幅が 10Mbps の場合ピア平均スループット累積分布関数

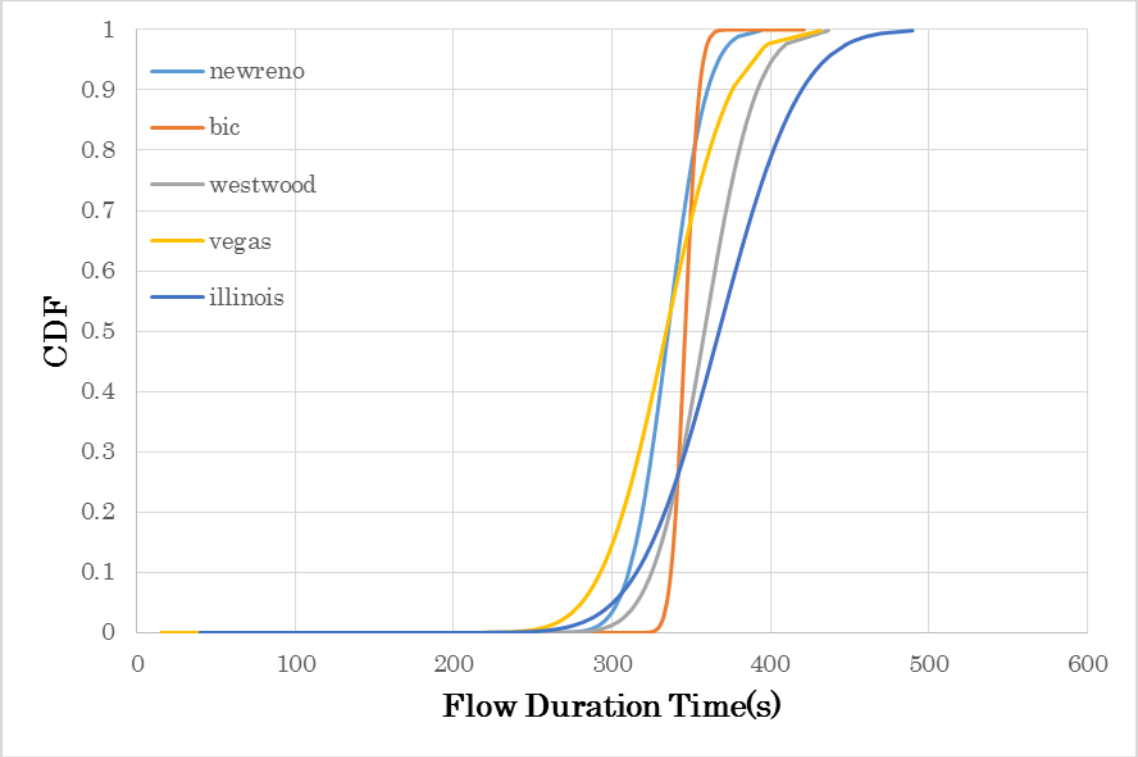

図 48　帯域幅が 100Mbps の場合 TCP フロー毎のフロー継続時間累積分布関数

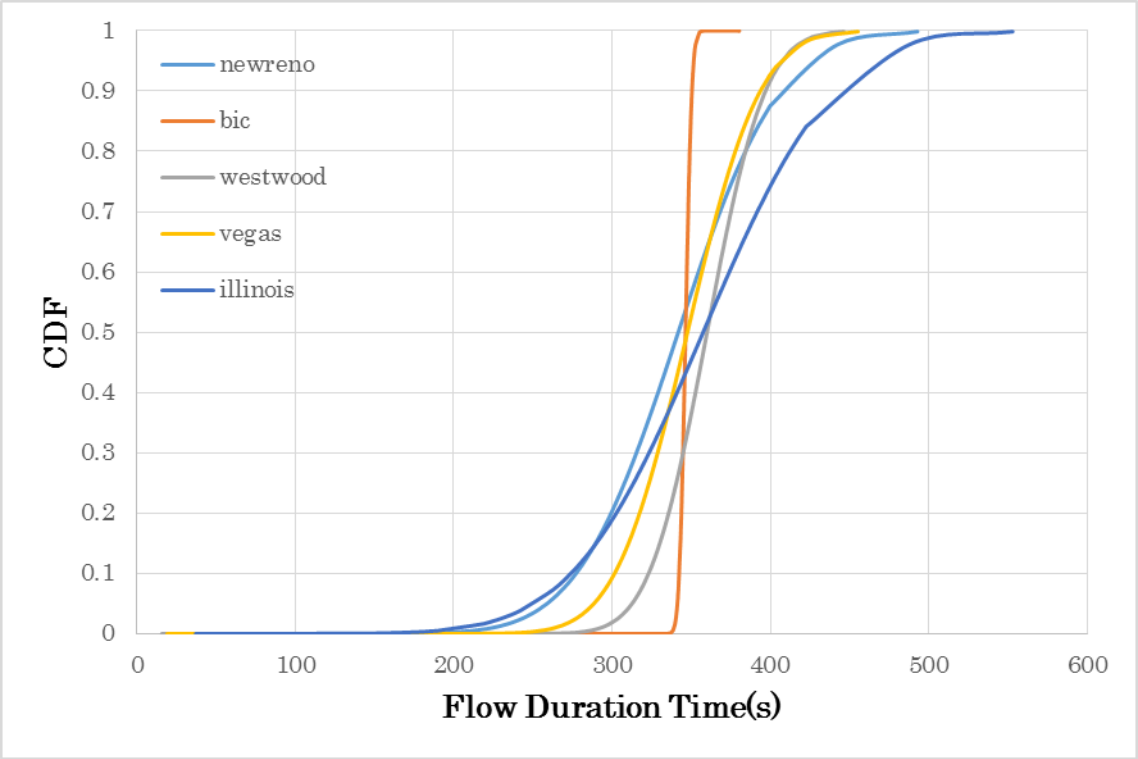

図 49　帯域幅が 10Mbps の場合 TCP フロー毎のフロー継続時間累積分布関数

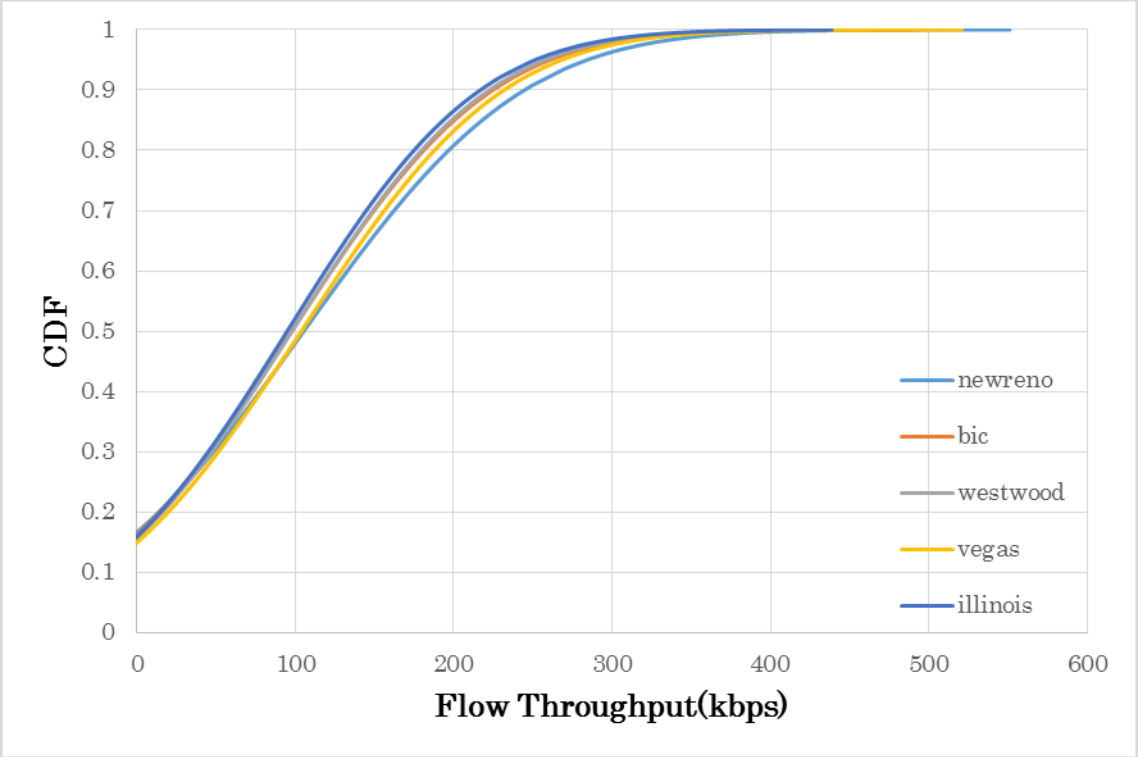

図 50　帯域幅が 100Mbps の場合 TCP フロー毎のフロースループット累積分布関数

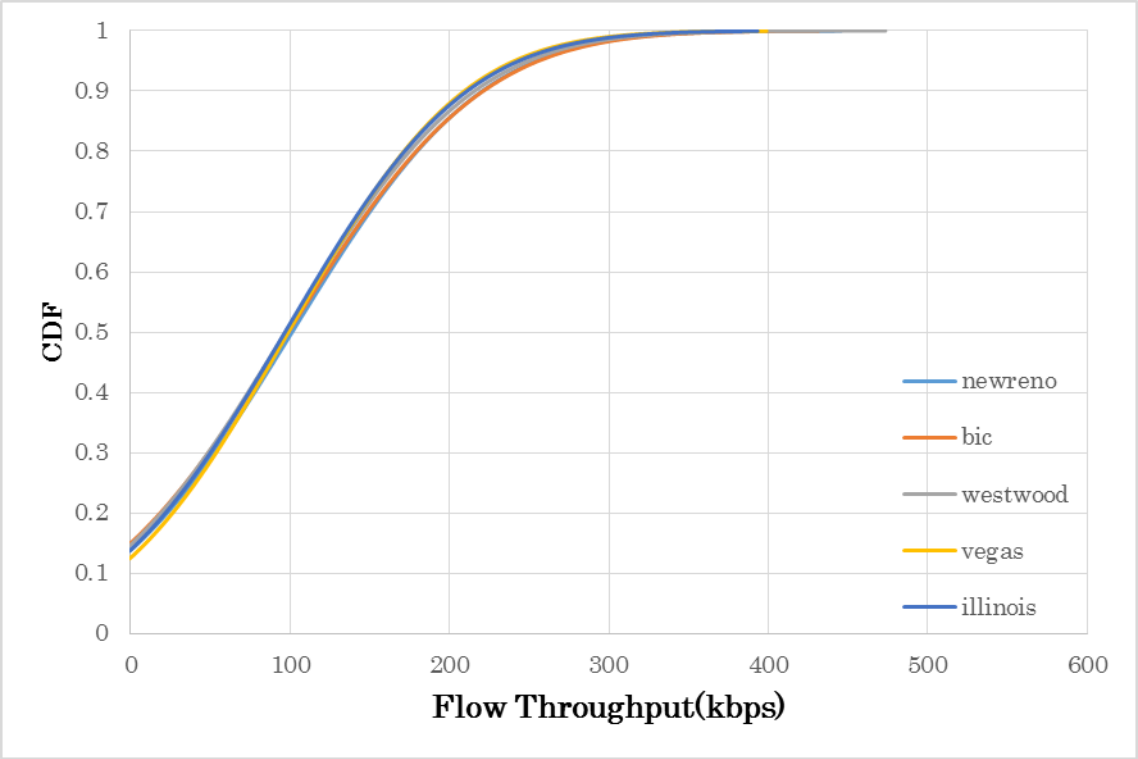

図 51　帯域幅が 10Mbps の場合 TCP フロー毎のフロースループット累積分布関数

4.3 非チョーク数が異なる場合

本実験では、シーダとリーチャの非チョーク数を変更し、分析を行う。3 章で述べた基本トポロジーのシーダとリーチャの非チョーク数はそれぞれ 3 と 10 に設定する。

4.3.1 ローカライゼーション度合い測定の結果

図 52 と図 53 は非チョーク数が 3 と 10 の場合のローカライゼーション度合いの累積分布関数を示す。図 54 と図 55 は非チョーク数が 3 と 10 の場合のダウンロードしたピースの平均ホップ数の累積分布関数を示す。非チョーク数が 10 の場合、TCP Illinois の性能が改善している。図 56 から図 65 に示された各ピアのピース取得状況のマトリックスを見ると、非チョーク数が 10 の場合、TCP Illinois はシーダから非チョークされたピア数が増え、AS 内のピース数が多くなる。そして、シーダから非チョークされてないピアは AS 外からダウンロードしたピースを非チョーク数が 3 の場合より積極的にダウンロードしたため、ローカライゼーション度合いが高く、ダウンロードしたピースの平均ホップ数も小さくなる。

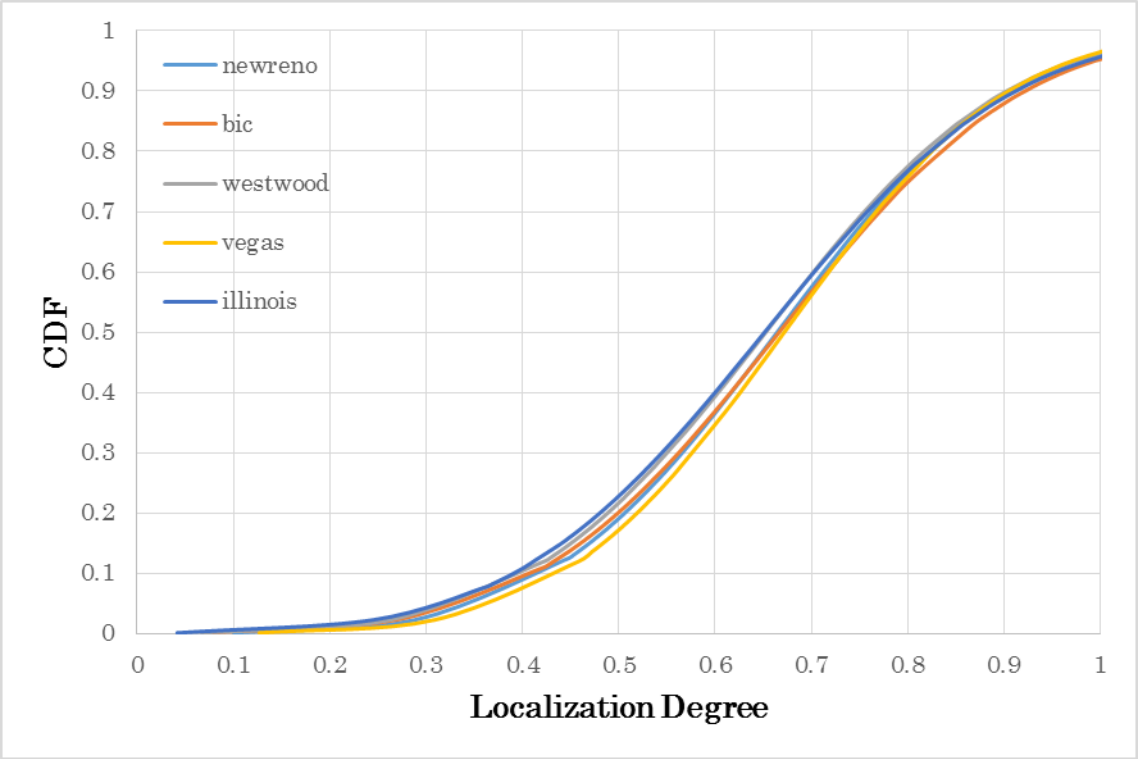

図 52　非チョーク数 3 の場合ローカライゼーション度合い累積分布関数

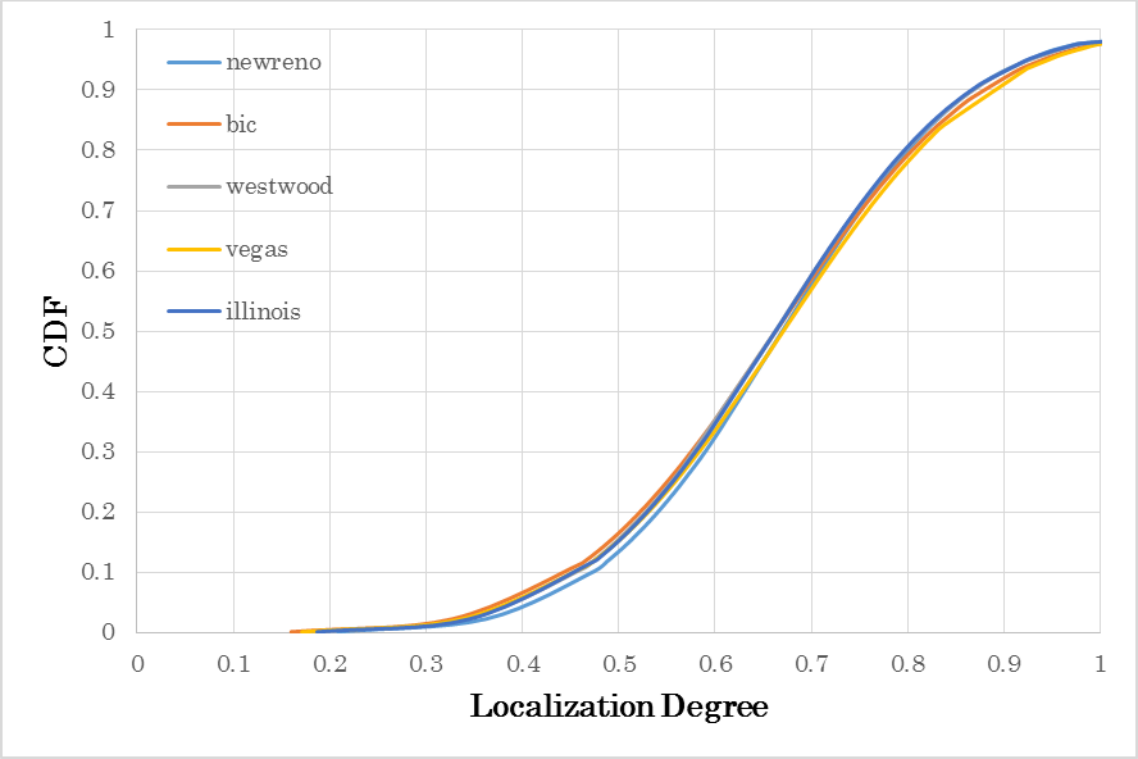

図 53　非チョーク数 10 の場合ローカライゼーション度合い累積分布関数

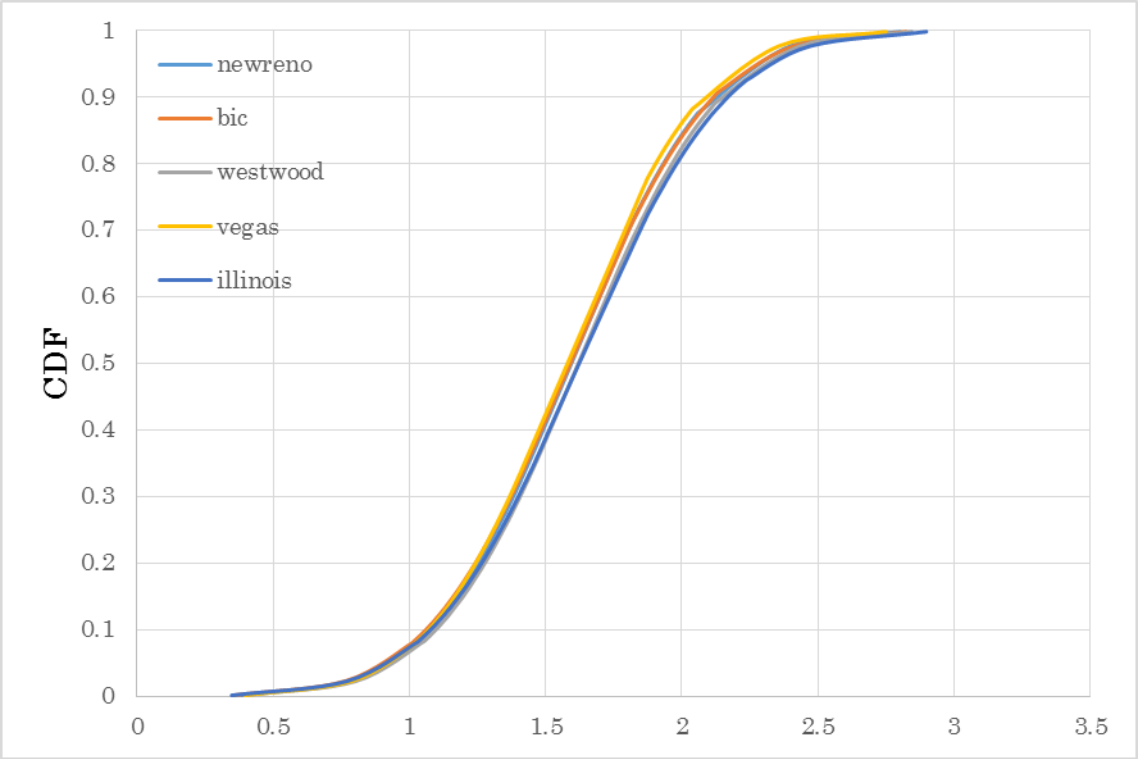

図 54　非チョーク数 3 の場合ダウンロードしたピースの平均ホップ数累積分布関数

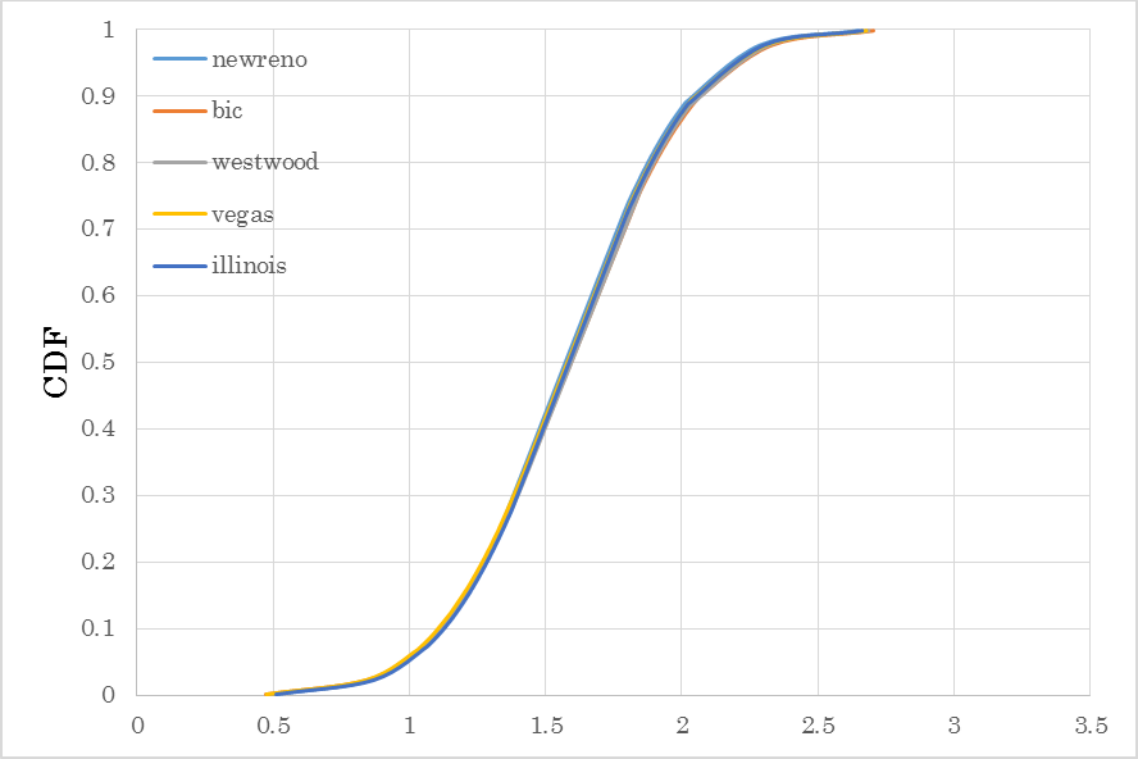

図 55　非チョーク数 10 の場合ダウンロードしたピースの平均ホップ数累積分布関数

図 56 から図 65 は非チョーク数が 3 と 10 の場合、TCP 輻輳制御アルゴリズムごとの各ピアのピース取得状況を示している。縦軸は送信側のピアの IP アドレスであり、横軸は受信側のピア

の IP アドレスである。横軸の IP アドレスの”10.2.0”は省略する。IP アドレスが”10.2.0.41”のピアはシーダである。表の数値は各ピアからダウンロードしたピース数を示している。紫色の部分は AS 内ピアからダウンロードしたピース数を表し、緑色は AS 外からの取得ピース数を表している。

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	15	45	0	35	25	1	30	0	333	611	0	492	0	209	0	248
10.2.0.43		106	116	109	108	122	116	113	6	0	3	5	1	1	2	4
10.2.0.45	114		120	113	101	113	123	99	18	18	18	20	16	19	19	16
10.2.0.47	112	110		106	102	110	115	112	2	0	4	4	4	0	2	3
10.2.0.49	125	126	139		128	130	116	129	12	14	19	8	14	16	10	13
10.2.0.51	134	111	140	120		135	132	136	7	5	9	7	9	8	9	7
10.2.0.53	120	108	131	122	118		120	118	2	0	1	1	3	1	1	0
10.2.0.55	82	94	93	88	89	100		99	14	16	12	15	16	15	13	15
10.2.0.57	82	77	86	78	91	79	76		4	0	2	0	1	0	0	0
10.2.0.59	118	126	71	96	88	97	98	112		312	297	315	252	303	215	217
10.2.0.61	101	111	98	109	95	93	104	108	561		578	540	498	561	609	436
10.2.0.63	91	101	100	100	106	97	107	104	0	8		0	34	3	42	42
10.2.0.65	95	104	102	103	106	91	96	94	427	409	434		456	430	440	450
10.2.0.67	98	94	99	95	119	112	101	95	1	1	1	0		1	1	2
10.2.0.69	112	95	90	96	105	101	91	94	4	2	2	0	4		4	19
10.2.0.71	114	100	101	102	96	104	94	99	5	0	2	2	65	10		110
10.2.0.73	106	94	96	96	98	104	93	93	193	179	204	171	209	208	206	

図 56　非チョーク数が 3 の場合 TCP New Reno の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	553	481	55	102	0	2	4	516	0	21	0	36	171	0	20	39
10.2.0.43		470	431	390	279	198	191	488	98	91	59	70	68	66	103	64
10.2.0.45	394		412	442	370	253	366	413	93	148	91	70	41	90	89	67
10.2.0.47	44	44		0	71	164	136	0	111	113	124	98	78	85	91	72
10.2.0.49	92	97	109		188	240	256	98	94	102	102	122	95	80	81	127
10.2.0.51	0	0	4	57		27	28	2	105	110	86	110	105	85	91	123
10.2.0.53	0	0	2	15	2		4	14	83	110	117	103	119	98	100	111
10.2.0.55	0	0	11	17	15	28		10	99	93	119	100	103	130	85	114
10.2.0.57	452	446	510	517	617	629	558		101	98	115	94	98	116	117	91
10.2.0.59	4	6	1	0	4	2	0	4		61	78	77	82	78	66	62
10.2.0.61	3	2	4	9	8	2	6	6	141		131	142	160	136	142	144
10.2.0.63	2	2	13	2	0	7	9	7	115	118		95	123	114	121	119
10.2.0.65	14	1	9	14	12	9	3	0	116	102	109		117	134	128	151
10.2.0.67	5	0	2	6	4	5	1	8	110	109	109	101		128	119	105
10.2.0.69	9	2	0	1	3	0	0	8	90	86	94	94	90		83	93
10.2.0.71	8	7	9	0	1	9	8	12	127	122	121	126	164	141		136
10.2.0.73	0	22	8	1	13	4	8	2	149	174	151	171	154	146	169	

図 57　非チョーク数が 3 の場合 BIC-TCP の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	546	532	0	0	447	0	158	3	71	0	20	0	23	24	32	16
10.2.0.43		499	468	262	510	524	247	324	98	108	116	75	108	80	95	110
10.2.0.45	462		424	413	436	472	533	548	104	108	102	82	89	82	97	98
10.2.0.47	0	0		18	0	24	31	23	105	103	104	100	83	90	96	94
10.2.0.49	2	0	1		0	1	0	1	96	97	104	118	88	92	108	100
10.2.0.51	381	377	403	429		358	388	386	101	96	86	111	92	114	102	99
10.2.0.53	2	0	51	148	2		141	105	92	87	99	110	117	102	95	115
10.2.0.55	107	90	145	204	105	120		116	89	100	100	99	109	116	90	94
10.2.0.57	0	0	9	30	1	1	6		93	96	95	93	98	104	116	95
10.2.0.59	36	21	25	23	24	20	22	21		145	132	144	141	139	126	128
10.2.0.61	9	15	12	5	1	0	4	8	59		45	63	54	60	59	59
10.2.0.63	16	15	15	10	9	10	15	14	107	103		117	117	114	116	120
10.2.0.65	4	0	4	7	7	6	3	4	80	83	82		79	88	85	82
10.2.0.67	5	8	8	5	3	14	9	9	93	97	113	89		96	92	87
10.2.0.69	5	10	8	10	17	13	12	12	118	110	111	107	127		125	114
10.2.0.71	8	10	9	12	12	13	7	13	140	137	140	151	135	144		155
10.2.0.73	4	5	3	1	2	2	3	5	121	97	119	109	108	112	124	

図 58　非チョーク数が 3 の場合 TCP Westwood の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	22	44	8	0	23	107	22	0	511	431	215	201	0	168	376	36
10.2.0.43		126	135	139	138	135	134	123	0	0	0	0	0	2	1	0
10.2.0.45	148		165	171	157	166	160	159	2	1	0	9	0	0	0	0
10.2.0.47	103	119		111	117	107	115	116	1	0	0	0	0	1	0	0
10.2.0.49	113	118	113		107	116	104	124	0	0	1	1	1	1	8	0
10.2.0.51	98	98	111	108		109	105	107	0	0	0	0	2	0	0	0
10.2.0.53	92	112	103	105	105		102	101	0	1	0	8	8	0	0	0
10.2.0.55	118	125	122	136	133	130		117	8	0	0	0	1	0	0	0
10.2.0.57	129	122	126	123	110	107	100		0	0	3	1	0	1	2	1
10.2.0.59	79	81	99	65	67	75	76	92		369	415	383	314	432	390	368
10.2.0.61	87	89	103	77	101	104	69	92	185		224	241	269	234	234	182
10.2.0.63	80	78	99	83	87	102	113	100	294	214		228	206	237	187	279
10.2.0.65	98	111	91	85	89	95	108	98	192	130	180		178	172	162	234
10.2.0.67	87	102	84	86	92	94	101	100	15	12	28	4		20	25	24
10.2.0.69	113	101	86	94	70	80	93	91	7	54	55	48	55		55	62
10.2.0.71	114	92	91	105	82	93	82	96	344	340	335	346	375	374		381
10.2.0.73	107	93	93	106	107	100	98	103	9	15	114	105	168	94	137	

図 59　非チョーク数が 3 の場合 TCP Vegas の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	44	26	111	28	23	0	39	0	332	603	406	4	59	286	0	0
10.2.0.43		126	92	129	128	106	122	108	12	4	10	5	17	10	16	16
10.2.0.45	119		135	141	132	124	121	135	13	20	13	19	12	19	14	13
10.2.0.47	96	129		111	121	107	116	114	1	2	1	5	4	4	1	2
10.2.0.49	110	121	120		117	119	103	126	3	9	4	7	7	7	8	8
10.2.0.51	98	98	105	105		108	97	103	5	8	6	5	6	2	7	4
10.2.0.53	81	76	88	73	76		86	73	3	1	2	2	1	2	2	5
10.2.0.55	133	138	141	142	131	139		133	15	16	15	13	12	13	12	9
10.2.0.57	77	81	78	83	82	81	81		0	0	0	0	0	0	0	0
10.2.0.59	100	104	95	110	87	89	99	90		320	327	342	302	281	288	0
10.2.0.61	108	120	91	131	118	117	126	112	479		513	436	418	381	310	471
10.2.0.63	106	105	93	110	113	113	105	107	389	328		403	388	343	381	548
10.2.0.65	100	99	96	102	101	105	104	103	1	3	0		31	45	11	110
10.2.0.67	100	88	128	117	110	102	125	107	67	55	58	69		65	113	82
10.2.0.69	107	91	119	91	96	95	111	116	250	216	213	273	260		279	289
10.2.0.71	105	103	104	103	111	105	101	100	3	5	5	2	4	1		25
10.2.0.73	110	111	107	101	100	99	116	93	3	0	4	3	61	123	146	

図 60　非チョーク数が 3 の場合 TCP Illinois の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	31	14	19	0	34	19	33	58	284	276	283	227	258	98	265	282
10.2.0.43		102	99	89	93	93	99	98	0	3	3	3	5	12	5	12
10.2.0.45	89		83	86	83	84	85	85	0	1	13	7	7	3	2	3
10.2.0.47	142	143		132	118	119	128	117	8	8	2	2	3	5	3	3
10.2.0.49	94	126	103		99	93	107	88	8	3	3	1	1	3	0	0
10.2.0.51	102	111	101	105		109	102	103	16	13	10	14	13	12	10	10
10.2.0.53	119	129	119	110	124		115	123	1	0	0	0	1	5	1	1
10.2.0.55	128	140	122	128	123	133		133	6	15	10	13	15	14	17	16
10.2.0.57	165	189	166	171	160	163	165		21	18	18	23	19	15	21	16
10.2.0.59	74	66	98	105	85	93	87	107		208	208	214	206	156	210	215
10.2.0.61	87	2	102	103	93	90	91	96	204		208	207	206	178	198	204
10.2.0.63	79	98	92	94	96	101	86	92	216	219		220	221	249	215	219
10.2.0.65	82	88	96	86	87	82	86	84	162	160	161		162	196	160	162
10.2.0.67	90	89	91	85	109	99	88	95	188	191	191	189		207	195	184
10.2.0.69	99	85	95	80	91	92	87	110	47	45	47	47	45		58	48
10.2.0.71	99	89	89	94	86	96	88	95	203	201	205	199	201	207		202
10.2.0.73	91	90	90	96	81	100	103	89	211	215	214	212	211	225	212	

図 61　非チョーク数が 10 の場合 TCP New Reno の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	47	19	39	18	46	0	30	20	324	279	340	223	271	188	17	287
10.2.0.43		84	118	123	118	110	106	126	9	10	13	17	10	12	15	16
10.2.0.45	90		115	108	112	117	101	117	0	0	2	0	1	0	0	0
10.2.0.47	133	141		144	136	131	134	134	2	6	4	5	7	4	5	4
10.2.0.49	107	112	98		98	97	104	90	0	0	0	0	1	3	2	2
10.2.0.51	154	192	152	152		148	151	137	12	12	10	9	12	15	10	12
10.2.0.53	94	129	112	117	102		107	111	4	7	2	0	4	2	2	1
10.2.0.55	115	121	121	135	122	126		134	8	4	9	6	6	4	7	4
10.2.0.57	102	112	90	91	88	97	90		1	1	2	1	2	0	4	1
10.2.0.59	90	67	68	80	91	101	96	80		260	256	267	250	187	222	208
10.2.0.61	82	68	94	76	90	96	96	89	220		208	210	211	252	204	195
10.2.0.63	85	81	96	94	92	91	98	87	270	268		258	243	277	313	288
10.2.0.65	80	87	94	83	99	87	96	85	160	159	158		175	164	175	176
10.2.0.67	86	86	96	87	86	92	94	88	216	211	213	216		217	224	219
10.2.0.69	116	88	94	83	95	88	91	88	111	120	115	114	122		131	138
10.2.0.71	98	88	90	91	100	92	80	79	20	17	20	20	36	24		25
10.2.0.73	94	91	87	84	90	92	91	98	218	220	227	225	226	225	238	

図 62　非チョーク数が 10 の場合 BIC-TCP の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	46	37	29	25	35	19	85	34	328	315	398	0	285	213	61	303
10.2.0.43		150	143	149	150	110	96	139	20	12	22	10	9	24	2	2
10.2.0.45	156		164	158	149	161	158	152	15	16	14	6	15	9	7	19
10.2.0.47	87	80		78	77	80	83	74	4	6	9	3	10	6	2	8
10.2.0.49	107	109	115		88	104	87	91	16	16	19	14	16	12	1	15
10.2.0.51	103	93	103	86		97	104	99	5	7	3	8	7	7	10	7
10.2.0.53	80	71	71	81	77		84	74	0	4	0	1	1	0	14	1
10.2.0.55	72	113	110	113	106	114		114	0	0	0	6	3	0	5	1
10.2.0.57	167	159	153	152	166	170	178		4	7	1	7	1	9	12	12
10.2.0.59	84	105	98	102	104	84	90	102		238	248	256	255	258	252	267
10.2.0.61	99	92	83	95	86	89	94	110	230		230	241	231	235	248	247
10.2.0.63	90	96	92	85	91	84	112	95	326	327		346	328	326	298	300
10.2.0.65	95	86	78	79	94	90	110	87	0	0	0		1	0	10	13
10.2.0.67	98	99	88	90	88	92	92	102	213	216	215	230		213	249	207
10.2.0.69	113	117	84	95	85	92	96	103	146	146	147	163	151		161	154
10.2.0.71	77	77	75	76	77	92	92	111	22	24	29	33	24	27		24
10.2.0.73	91	92	87	87	100	85	87	81	246	245	242	253	242	235	240	

図 63　非チョーク数が 10 の場合 TCP Westwood の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	52	32	19	28	34	34	19	84	358	366	270	354	247	292	19	0
10.2.0.43		130	121	135	124	125	129	135	30	12	18	18	6	9	0	1
10.2.0.45	123		130	126	119	127	146	134	2	8	7	9	7	12	18	12
10.2.0.47	120	116		116	120	110	141	153	0	5	2	5	7	1	2	1
10.2.0.49	125	114	107		113	112	117	108	5	5	5	7	10	6	7	10
10.2.0.51	145	135	119	122		124	136	163	3	4	7	6	7	6	6	7
10.2.0.53	107	112	107	113	109		127	153	9	12	10	3	7	9	9	14
10.2.0.55	89	80	87	83	83	79		88	2	1	7	0	0	3	4	0
10.2.0.57	118	105	127	112	111	119	111		0	1	0	0	1	0	0	0
10.2.0.59	97	112	102	92	102	87	89	66		287	294	298	274	230	267	286
10.2.0.61	85	98	113	94	105	85	81	54	292		291	284	281	255	330	297
10.2.0.63	99	93	96	93	97	102	84	66	211	212		195	230	249	238	201
10.2.0.65	96	93	98	91	108	102	3	66	278	284	281		283	271	282	305
10.2.0.67	81	83	98	96	93	98	77	98	174	175	174	182		192	191	202
10.2.0.69	85	86	88	80	79	83	106	95	207	203	210	211	208		217	225
10.2.0.71	67	92	80	85	81	94	94	99	0	0	0	0	4	29		13
10.2.0.73	76	88	75	98	88	89	84	88	2	0	1	2	3	19	6	

図 64　非チョーク数が 10 の場合 TCP Vegas の各ピアのピース取得状況

reciever / sender	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73
10.2.0.41	347	258	242	319	304	299	0	36	41	71	61	41	20	112	99	51
10.2.0.43		268	268	284	282	253	185	265	106	101	87	87	63	76	83	87
10.2.0.45	188		195	188	189	174	185	197	105	78	98	98	105	103	62	86
10.2.0.47	210	209		208	210	203	175	201	109	95	89	88	92	101	95	105
10.2.0.49	243	249	255		237	278	179	219	85	87	91	85	99	86	112	101
10.2.0.51	233	237	248	226		250	241	207	95	98	95	122	103	102	102	103
10.2.0.53	232	235	232	235	232		463	224	96	98	86	107	101	90	103	91
10.2.0.55	0	0	1	0	0	0		6	95	83	98	101	101	86	95	86
10.2.0.57	6	3	7	5	5	3	35		70	92	88	91	87	81	90	79
10.2.0.59	29	24	22	18	16	19	22	40		116	119	119	120	122	134	114
10.2.0.61	38	29	29	28	30	28	22	32	130		127	108	126	135	129	119
10.2.0.63	24	21	15	23	29	22	20	27	119	120		141	117	127	126	124
10.2.0.65	3	9	14	15	16	17	21	37	137	139	142		143	143	136	133
10.2.0.67	0	15	13	6	7	7	13	15	114	112	107	106		118	111	117
10.2.0.69	0	4	8	6	2	1	3	14	66	78	69	69	71		75	67
10.2.0.71	0	0	2	5	3	4	3	24	101	112	104	105	109	97		103
10.2.0.73	19	13	19	18	19	19	20	28	97	94	107	100	112	99	112	

図 65　非チョーク数が 10 の場合 TCP Illinois の各ピアのピース取得状況

4.3.2 ピア平均スループット測定の結果

図 66 は非チョーク数が異なる場合 TCP 輻輳制御アルゴリズム毎の再送回数を示す。図 67 と図 68 は非チョーク数が 3 と 10 の場合のスループット累積分布関数を示す。図 69 と図 70 は非チョーク数が 3 と 10 の場合の TCP フロー毎の TCP フロー継続時間の累積分布関数を示す。図 71 と図 72 は非チョーク数が 3 と 10 の場合の TCP フロー毎の TCP フロースループットの累積分布関数を示す。図 66 に示されたように非チョーク数が 10 の場合、全ての TCP 輻輳制御アルゴリズムの再送回数が少なくなる。しかし、ピア平均スループットは TCP Vegas のスループットがやや速くなる。非チョーク数が 10 の場合、TCP New Reno は再送回数が少なくなるが、スループットが速い TCP フローがなくなり、ピアの平均スループットが遅くなる。

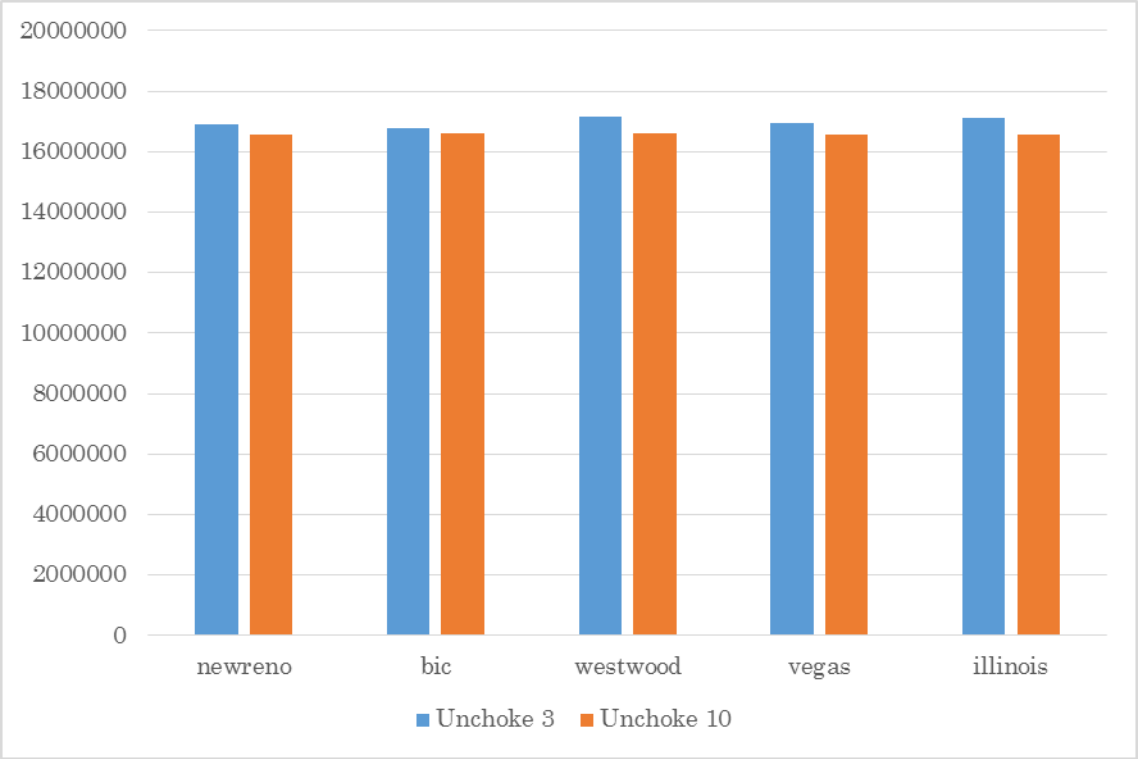

図 66　非チョーク数が異なる場合 TCP 輻輳制御アルゴリズム毎の再送回数

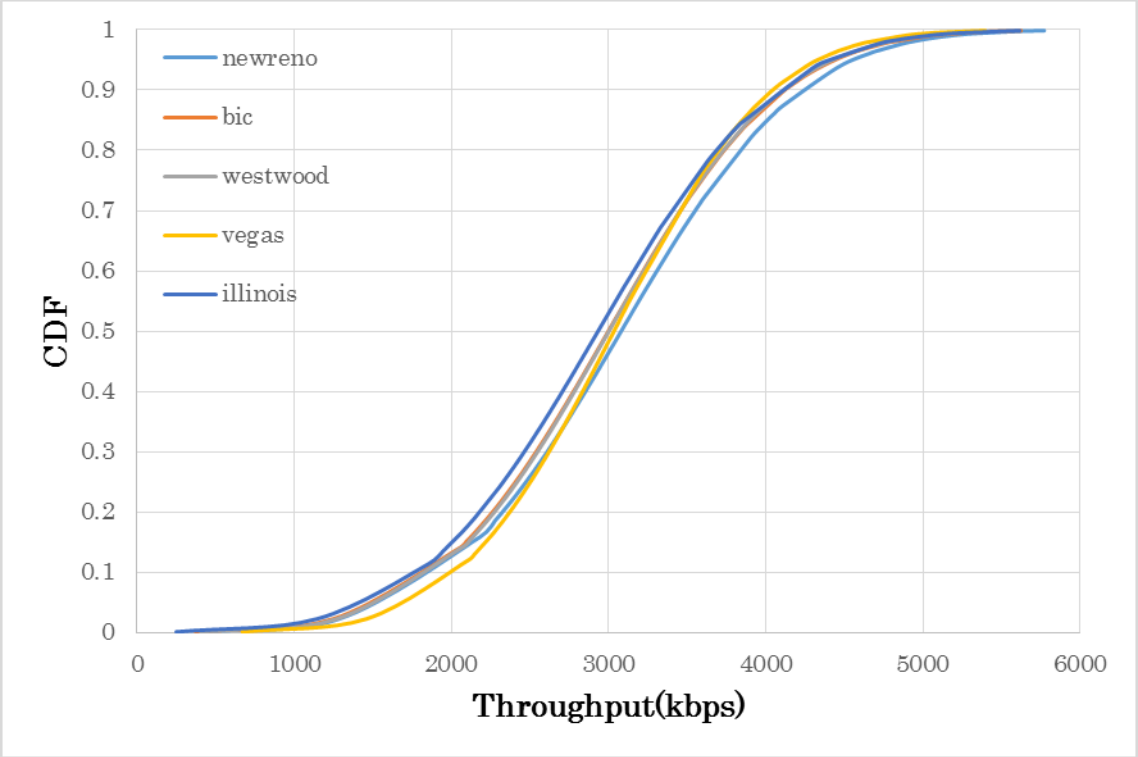

図 67　非チョーク数 3 の場合ピア平均スループット累積分布関数

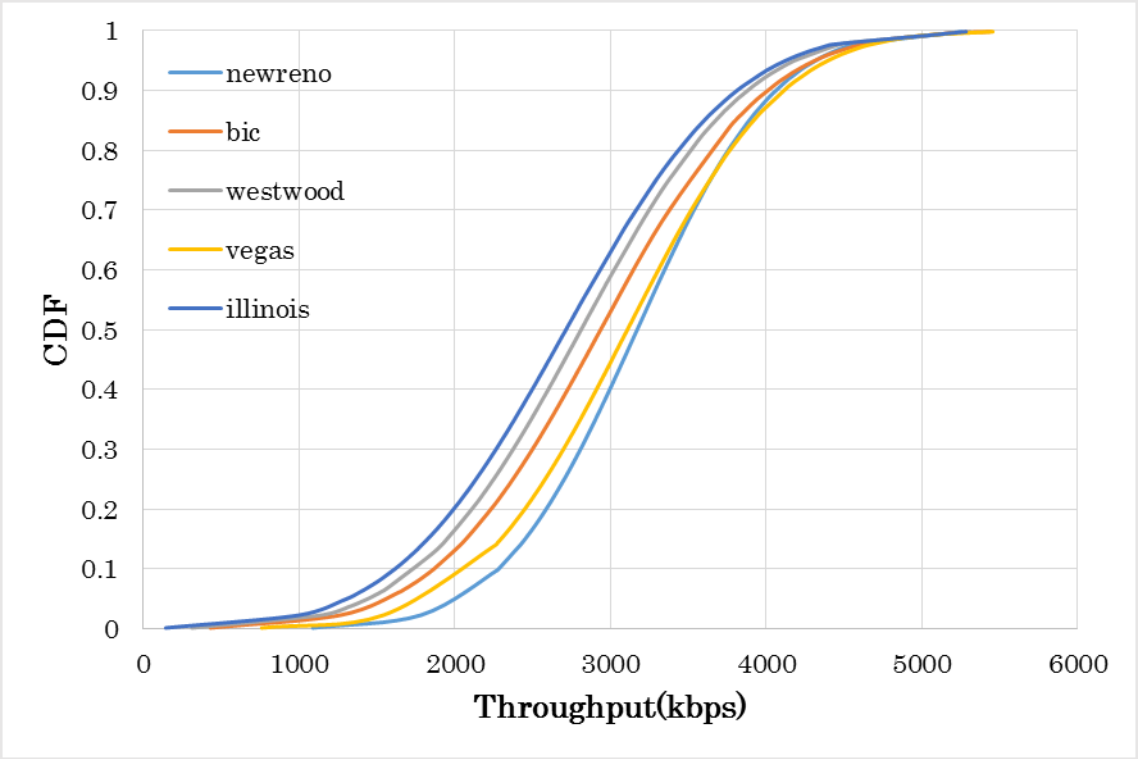

図 68　非チョーク数 10 の場合ピア平均スループット累積分布関数

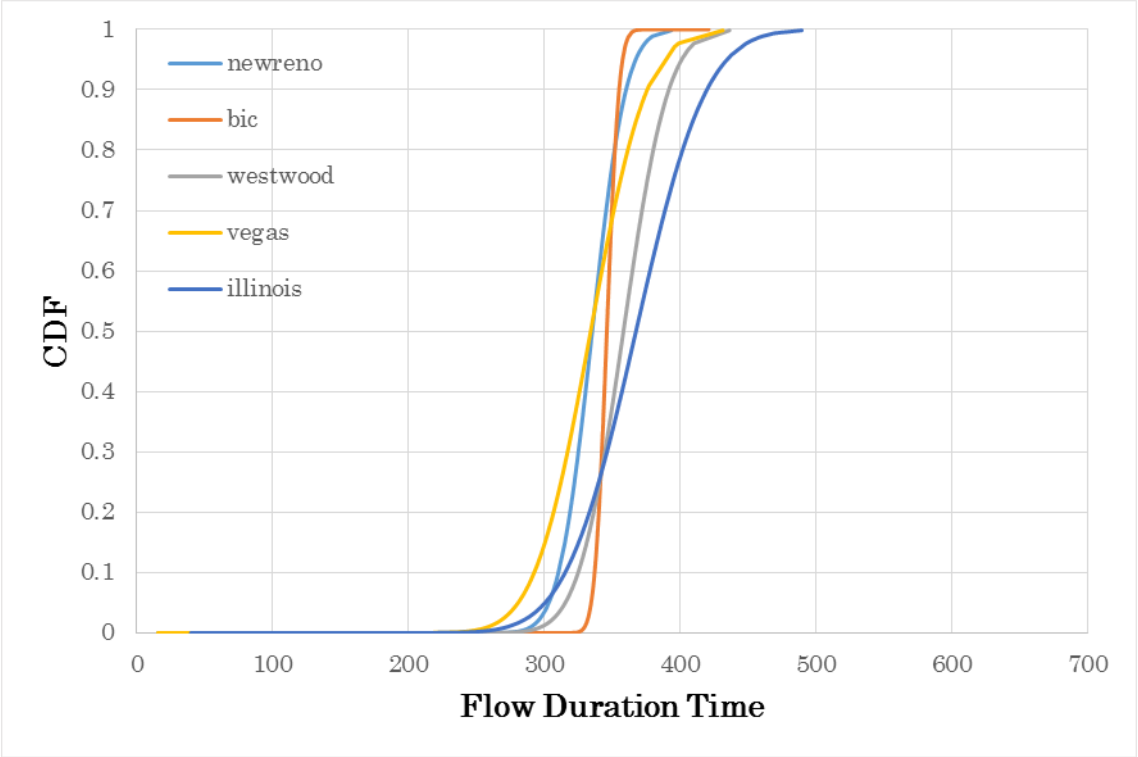

図 69　非チョーク数が 3 の場合 TCP フロー毎のフロー継続時間累積分布関数

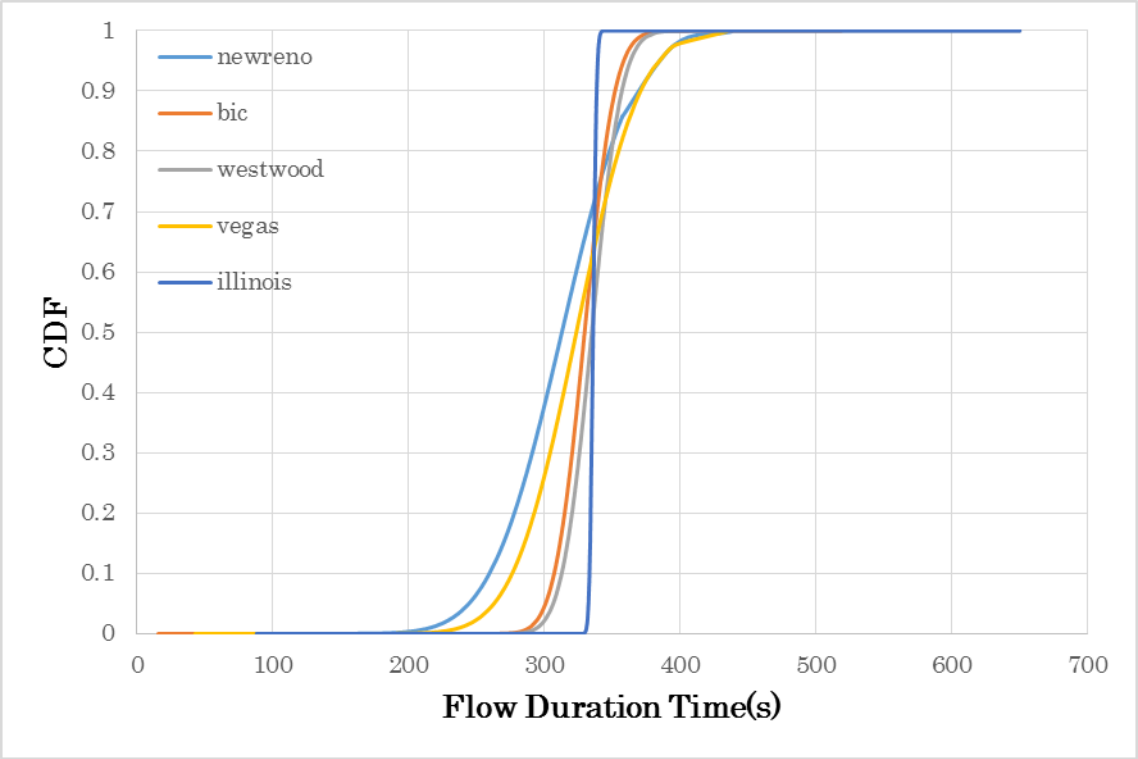

図 70　非チョーク数が 10 の場合 TCP フロー毎のフロー継続時間累積分布関数

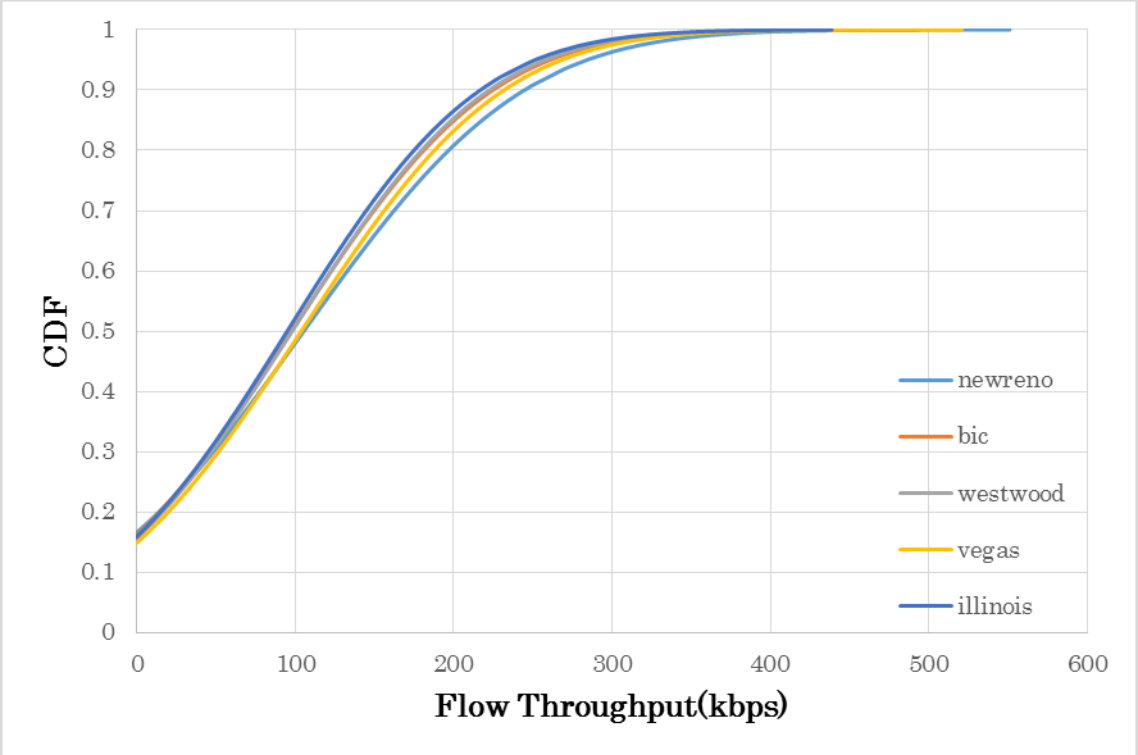

図 71　非チョーク数が 3 の場合 TCP フロー毎のフロースループット累積分布関数

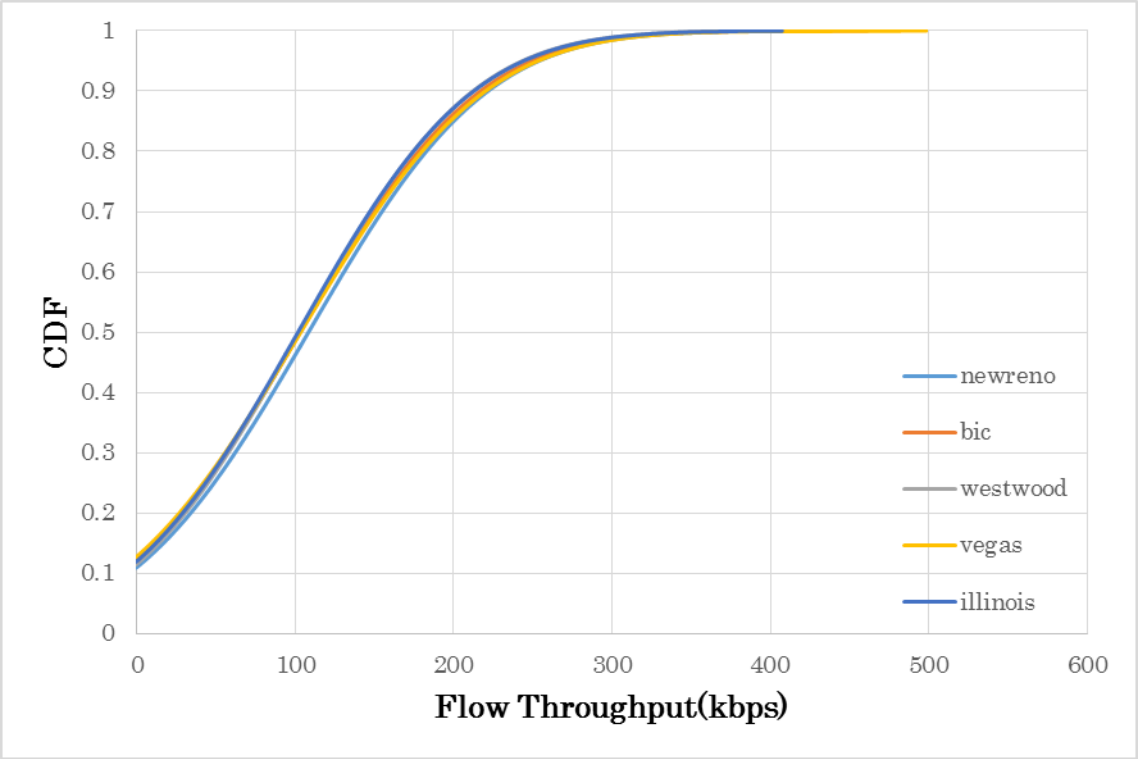

図 72　非チョーク数が 10 の場合 TCP フロー毎のフロースループット累積分布関数

第５章　結論

5.1 まとめ

本論文では、BitTorrent において、トポロジー側のパラメーターと BitTorrent 側のパラメーターを変更し、TCP 輻輳制御アルゴリズムごとに性能を評価した。TCP 輻輳制御アルゴリズムの違いによる BitTorrent の性能を調べるために各ピアのスループットの平均スループットを調査した。また、AS 間のトラヒック問題を考察するために、ピアを複数の AS に設定し、ピア間のピース選択状況を調べるために、ローカライゼーション度合いを調査した。

伝搬遅延を 1ms と 5ms に設定した場合、TCP Vegas は最初にシーダから非チョークされてないピアは、AS 外から持ってきたピースを他の TCP 輻輳制御アルゴリズムより AS 内で積極的にダウンロードするため、ローカライゼーション度合いも高くなる。伝搬遅延が 5ms の場合 BIC-TCP はシーダから大量のピースをダウンロードしたピア数が少なくなり、その AS 内のピース数が少なくなったため、ローカライゼーション度合いが悪くなる。また、伝搬遅延が 5ms の場合、BIC-TCP の再送回数が多くなるため、BIC-TCP の TCP フロー継続時間が長くなり、スループットが遅くなる。

帯域幅が 100Mbps の場合 TCP Vegas がローカライズされやすい。帯域幅が 10Mbps の場合 TCP Westwood は、シーダから選ばれたピアの AS において、ピース数が多いため、AS 内から積極的にピースをダウンロードするが、ピースが少ない AS のピアは AS 内と AS 外の帯域幅条件が同一になるため、帯域幅が 100Mbps の場合より、AS 外からのダウンロードが増え、ローカライゼーション度合いが悪くなる。また、帯域幅が 10Mbps の場合、BIC-TCP の再送回数がやや少なくなり、TCP フロー毎の継続時間も短くなり、帯域幅が 100Mbps の場合よりスループットが少し速くなる。

非チョーク数が 10 の場合、TCP Illinois はシーダから非チョークされたピア数が増え、AS 内のピース数が多くなる。そして、シーダから非チョークされてないピアは AS 外からダウンロードしたピースを非チョーク数が 3 の場合より積極的にダウンロードしたため、ローカライゼーション度合いが高くなる。また、非チョーク数が 10 の場合、全ての TCP 輻輳制御アルゴリズムの再送回数が少なくなるが、ピア平均スループットは TCP Vegas のスループットがやや速くなる。非チョーク数が 10 の場合、TCP New Reno は再送回数が少なくなるが、スループットが速い TCP フローがなくなる。

5.2 今後の課題

今後の課題として一つ目は、本実験の結果で、非チョーク数が 10 の場合、各 TCP 輻輳制御アルゴリズムの再送回数が少なくなったが、スループットが遅くなる原因をさらに分析して明ら

かにする必要がある。二つ目は、ピアのスウォーム参加タイミングを、2 つ AS を一定時間の間隔で交互に参加させたが、シーだが 1 つ AS のピアを非チョークする現象を解決する必要がある。3 つ目は本実験の結果に TCP 輻輳制御アルゴリズムのどの制御が効果的であるのかを議論する必要がある。4 つ目は Tit-for-tat 戦略のパラメーターを変更して評価を行う。

そして、本論文では、ピアを 2 つ AS に設定したトポロジーになる。更に、詳細な分析をするためには、AS 数を増やし、より複雑なネットワークトポロジーで、BitTorrent のピア間のピース選択状況を調べ、TCP 輻輳制御アルゴリズムごとにその傾向を考察する必要がある。また、複数の TCP が混雑している場合、各 TCP の競争性を分析する必要がある。そして、ns-3 で NSC を用いると、利用できる TCP 輻輳制御アルゴリズムが多数あるため、本論文で扱った TCP 輻輳制御アルゴリズム以外の TCP 輻輳制御アルゴリズムを考察できる。

謝辞

本研究を進めるにあたり、多くの方のご指導とご助言を賜りました。ここで深く感謝の意を表します。

本研究の主任指導教員である電気通信大学大学院情報システム学研究科ネットワークアーキテクチャ学講座の大坐畠智准教授には、研究方針に関して多大なご指導とご助言を頂き、充実な研究活動ができる環境を提供して頂きまして、心より感謝致します。

電気通信大学大学院情報システム学研究科ネットワークアーキテクチャ学講座の加藤聰彦教授、策力木格助教と山本嶺助教には、講座ゼミを通して多くのご助言を頂き、様々な問題点を解決して頂きました。

また、指導教員である電気通信大学大学院情報システム学研究科ネットワーク基礎学講座の長岡浩司教授にも、ご指導とご助言を頂き、深く感謝致します。

電気通信大学大学院情報システム学研究科ネットワークアーキテクチャ学講座の皆様、研究生活で大変お世話になりました。

以上、研究を進めるにあたり、多くの方に大変お世話になりました。心から感謝を申し上げます。

最後に、就学の機会を与えてくれた家族に感謝致します。

参考文献

[1]. Sandvine, https://www.sandvine.com/ (参照：2015/01/23). (オンライン)

[2]. S. Yamamoto, and A. Nakao, “P2P Packet Cache Router for Network-wide Traffic Redundancy Elimination,” Proc. 2012 International Conference on Computing, Networking and Communications (ICNC), pp. 830–834, 2012.

[3]. BitTorrent Still King of P2P Traffic, http://torrentfreak.com/BitTorrent-still-king-of-p2p-traffic-090218/ (参照：2015/01/09). (オンライン)

[4]. Bram Cohen, “Incentives Build Robustness in BitTorrent,” In Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems, pp.1-6, 2003.

[5]. 長谷川 剛, 村田 正幸, “高速・高遅延ネットワークのためのトランスポート層プロトコル,” 電子情報通信学会技術研究報告. IN2007, pp.41-46, 2007.

[6]. Lisong Xu, Khaled Harfoush, and Injong Rhee, “Binary Increase Congestion Control (BIC) for Fast Long-Distance Networks,” INFOCOM 2004, pp.2514-2524, 2004.

[7]. Claudio Casetti, Mario Gerla, Saverio Mascolo, M.Y.Sanadidi, and Ren Wang, “TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks,” Wireless Networks, Volume 8, Number 5, pp.467-479, 2002.

[8]. Luigi A. Grieco and Saverio Mascolo, “Performance Evaluation and Comparison of Westwood+, New Reno, and Vegas TCP Congestion Control,” ACM SIGCOMN Computer Communication Review Archive. Volume 34, Issue 2, pp.25-38, 2004.

[9]. Lawrence S. Brakmo and Larry L. Peterson, “TCP Vegas: End to End Congestion Avoidance on Global Internet,” IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, Volume 13, No.8, pp.1465-1480, 1995.

[10]. Shao Liu, Tamer Basar and R.Srikant, “TCP-Illinois: A Loss and Delay-Based Congestion Control Algorithm for High-Speed Networks,” Innovative Performance Evaluation Methodologies and Tools Vol.65, No.6, 7, pp.417-440, 2006.

[11]. Talal A. Edwin, Lin Guan, George Oikonomou, Iain Phillips, “Higher Order Delay Functions for Delay-Loss Based TCP Congestion Control,” Wireless Advanced, pp.1-6, 2010.

[12]. Dario Rossi, Claudio Testa, Silvio Valenti, “Yes, we LEDBAT: Playing with the new BitTorrent congestion control algorithm,” In Proceedings of the 11th Passive and Active Measurement Conference, pp.31-40, 2010.

[13]. Claudio Testa, Dario Rossi, “Delay-based congestion control: Flow vs. BitTorrent swarm perspectives,” Computer Networks, Volume 60, pp.115-128, 2014.

[14]. Luigi A. Grieco and Saverio Mascolo, “Performance Evaluation and Comparison of Westwood+, New Reno, and Vegas TCP Congestion Control,” ACM SIGCOMM Computer

Communication Review, Volume 34, No. 2, pp.25-38, 2004.
[15]. Bo Liu, Yi Cui, Yansheng Lu, and Yuan Xue, “Locality-Awareness in BitTorrent-like P2P Applications,” IEEE TRANSACTIONS ON MULTIMEDIA, Vol. 11, No.3, pp.361-370, 2009.
[16]. 近藤 大嗣, 中尾 彰宏, “BitTorrent における BITFIELD/HAVE メッセージのモニタリングによる網内レアピース検知手法,” 電子情報通信学会技術研究報告：信学技報 114(206), pp.73-78, 2014.
[17]. 韓 喆, “BitTorrent における TCP バージョンの違いによる影響のクロスレイヤ測定分析,” 電気通信大学大学院情報システム学研究科 修士論文, 2012
[18]. ns-3 simulator, http://www.nsnam.org/ (参照：2014/1/14). (オンライン)
[19]. TCP models in ns-3, http://www.nsnam.org/docs/models/html/tpc.html (参照 : 2014/2/10). (オンライン)
[20]. Elias Weingartner, Rene Glebke, Martin Lang, and Klaus Wehrle, “Building a modular BitTorrent model for ns-3,” SIMUROOLS ’12 Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques, pp.337-344, 2012.
[21]. The BRITE Output Format, http://www.cs.bu.edu/brite/user_manual/node29.html (参照：2014/10/10). (オンライン)